Distributed canonical correlation analysis in wireless sensor networks with application to distributed blind source separation

Alexander Bertrand, *Member, IEEE,* and Marc Moonen, *Fellow, IEEE*

***Abstract*—Canonical correlation analysis (CCA) is a widely-used data analysis tool that allows to assess the correlation between two distinct sets of signals. It computes optimal linear combinations of the signals in both sets such that the resulting signals are maximally correlated. The weight vectors defining these optimal linear combinations are referred to as 'principal CCA directions'. In addition to this particular type of data analysis, CCA is also often used as a blind source separation (BSS) technique, i.e., under certain assumptions, the principal CCA directions have certain demixing properties. In this paper, we propose a distributed CCA (DCCA) algorithm that can operate in wireless sensor networks (WSNs) with a fully-connected or a tree topology. The algorithm estimates the Q principal CCA directions from the sensor signal observations collected by the different nodes in the WSN, and extracts the corresponding sources. These network-wide principal CCA directions are estimated in a time-recursive fashion without explicitly constructing the corresponding network-wide correlation matrices, i.e., without the need for data centralization. Instead, each node locally computes smaller CCA problems, and only transmits compressed sensor signal observations (of dimension Q), which significantly reduces the bit rate over the wireless links of the WSN. We prove convergence and optimality of the DCCA algorithm, and we demonstrate its performance by means of numerical simulations in a blind source separation scenario.**

EDICS: SAM-MCHA Multichannel processing, SEN Signal Processing for Sensor Networks

***Index Terms*—Wireless sensor networks (WSNs), canonical correlation analysis, distributed estimation, blind source separation**

I. Introduction

Canonical correlation analysis (CCA) is a widely-used data analysis tool to assess the correlation between two distinct sets of data or signals [1], [2]. Basically, it looks for directions in the data with maximal cross-correlation, i.e., it computes optimal linear combinations of the signals in both sets such that the resulting signals are maximally correlated. It is strongly related to partial least squares (PLS) [3], which searches for directions with maximal covariance instead of maximal correlation. Both techniques can be viewed as extensions of principal component analysis (PCA) towards a two-set framework. Further extensions of CCA towards more than two data sets (multi-set CCA or M-CCA), have also been proposed [4], [5], but these are beyond the scope of this paper.

The CCA concept has become a widely-used signal processing (SP) tool, in particular in biomedical signal processing [5]–[13], but also in array processing [14], radar anti-jamming [15], speaker identification [16], SIMO and MIMO equalization [17], [18], and even for analyzing financial data [19]. It is noted that, due to its multi-set framework, CCA is often used for multi-modal signal analysis [7]–[9], [12], [16].

One important SP application of CCA is blind source separation (BSS) [5], [20]–[23]. This CCA-based BSS approach assumes that the hidden sources have a different autocorrelation structure, which is a valid assumption in many applications. For example, CCA-based BSS has been used to remove ocular artifacts [12], muscle artifacts [10], and ballistocardiographic artifacts [7] in electroencephalography (EEG) data. The technique fits in the family of second-order statistics (SOS) BSS techniques [24]–[26], and has a significantly lower computational complexity compared to higher-order statistics BSS techniques, such as independent component analysis [27] and fast implementations thereof [20].

In this paper, we consider the use of CCA to analyze data that is collected by a wireless sensor networks (WSNs), possibly in (slowly-varying) dynamic scenarios where the CCA directions may change over time (e.g. for adaptive BSS). Computing the network-wide CCA requires two matrix inversions and a (generalized) eigenvalue decomposition involving three network-wide correlation matrices that together capture the correlation between every possible sensor signal pair in the WSN. In principle, this would require to forward all the raw sensor signal observations to a fusion center (FC) where the network-wide correlation matrices can then be estimated. However, this usually results in a high bit rate over the wireless links, and it requires a high routing efficiency (in the case of partially-connected networks), and an FC with a large computational power.

To address these practical problems, we propose a distributed CCA (DCCA) algorithm, which avoids data centralization, i.e., it estimates the Q principal CCA directions

This work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven Research Council BOF/STG-14-005, CoE PFV/10/002 (OPTEC), the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office IUAP P7/23 (BESTCOM, 2012-2017), project iMinds Medical IT, Research Projects FWO nr. G.0763.12 'Wireless acoustic sensor networks for extended auditory communication', FWO nr. G.0931.14 'Design of distributed signal processing algorithms and scalable hardware platforms for energy-vs-performance adaptive wireless acoustic sensor networks', and HANDiCAMS. The project HANDiCAMS acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 323944. The scientific responsibility is assumed by its authors.

The authors are with KU Leuven, Department of Electrical Engineering (ESAT), Stadius Center for Dynamical Systems, Signal Processing and Data Analytics, Kasteelpark Arenberg 10, box 2446, 3001 Leuven, Belgium (e-mail: alexander.bertrand@esat.kuleuven.be, marc.moonen@esat.kuleuven.be). Alexander Bertrand is also with iMinds Medical IT.

without explicitly constructing the network-wide correlation matrices. Instead, each node only transmits Q-dimensional compressed observations of its sensor signals. The nodes then solve local CCA problems based on these compressed observations, i.e., based on correlation matrices with a much smaller dimension than in the centralized CCA. Hence, both the communication and computation cost are significantly reduced, be it at the cost of a slower tracking due to the iterative time-recursive nature of the DCCA algorithm.

For illustration purposes, and to make the generic problem statement more tangible, we will also briefly address how the DCCA algorithm can be used for distributed BSS in a WSN. The latter can be used, e.g., to remove artifacts in the signals recorded by wireless body area networks [28] or wireless EEG sensor networks [29]–[31], using the techniques in [7], [10], [12].

For the sake of an easy exposition, we will first describe the DCCA algorithm in fully-connected WSNs, i.e., where a signal broadcast by a node is collected by all other nodes in the network. Assuming that Q is smaller than the dimension of the per-node sensor observations, then the DCCA algorithm can provide a substantial reduction in processing and communication cost, although the per-node power consumption still depends on the total number of nodes in the network. We will then briefly explain how the DCCA algorithm can be generalized to partially-connected networks, using similar strategies as in [32], [33], where we assume that the network has been pruned to a tree topology to avoid feedback loops. The latter results in a fully scalable[1] DCCA algorithm with in-network data fusion, where the per-node processing cost and communication cost is independent of the number of nodes in the network.

The outline of the paper is as follows. In Section II, we briefly review CCA, and in Section III, we define the problem statement when applying CCA in WSNs. In Section IV, we derive the DCCA algorithm for fully-connected WSNs, and we prove its convergence to the centralized CCA solution. In Section V, we illustrate how this DCCA algorithm can be used as a distributed BSS algorithm. In section VI, we define the DCCA in WSNs with a tree topology. In Section VII, we demonstrate the performance of the DCCA algorithm by means of numerical simulations in a BSS scenario. Finally, conclusions are drawn in Section VIII.

II. Review of CCA

Consider an M-dimensional[2] stochastic signal $\mathbf{x}$ and an N-dimensional stochastic signal $\mathbf{y}$, each defining a (short-term) stationary and ergodic stochastic process. We define $\mathbf{x}[t]$ and $\mathbf{y}[t]$ as the t-th observation of $\mathbf{x}$ and $\mathbf{y}$, respectively, i.e., t denotes the sample index. Without loss of generality (w.l.o.g.), we assume that $\mathbf{x}$ and $\mathbf{y}$ are zero-mean, possibly requiring a mean subtraction pre-processing step on the set of observations.

In [1], CCA was introduced as a statistical method to find directions in the M- and N-dimensional space in which $\mathbf{x}$ and $\mathbf{y}$ are maximally correlated. To be more precise, CCA first identifies the M-dimensional vector $\hat{\mathbf{v}}_1$ and the N-dimensional vector $\hat{\mathbf{w}}_1$ such that the 1-dimensional stochastic signals $x_{v_1} = \hat{\mathbf{v}}_1^T\mathbf{x}$ and $y_{w_1} = \hat{\mathbf{w}}_1^T\mathbf{y}$ have maximum cross-correlation, i.e.,

$$(\hat{\mathbf{v}}_1, \hat{\mathbf{w}}_1) = \underset{(\mathbf{v}_1,\mathbf{w}_1)}{\arg\max} \frac{E\{\mathbf{v}_1^T\mathbf{x}\cdot\mathbf{y}^T\mathbf{w}_1\}}{\sqrt{E\{\mathbf{v}_1^T\mathbf{x}\cdot\mathbf{x}^T\mathbf{v}_1\}E\{\mathbf{w}_1^T\mathbf{y}\cdot\mathbf{y}^T\mathbf{w}_1\}}} \tag{1}$$

where $E\{\cdot\}$ denotes the expectation operator, superscript T denotes the transpose operator, and the notation $(\cdot\,,\cdot)$ denotes an ordered pair of variables. In the sequel, we write $\mathbf{R}_{xx}$, $\mathbf{R}_{yy}$, and $\mathbf{R}_{xy}$ to denote $E\{\mathbf{x}\mathbf{x}^T\}$, $E\{\mathbf{y}\mathbf{y}^T\}$, and $E\{\mathbf{x}\mathbf{y}^T\}$, respectively, such that (1) can be written as

$$(\hat{\mathbf{v}}_1, \hat{\mathbf{w}}_1) = \underset{(\mathbf{v}_1,\mathbf{w}_1)}{\arg\max} \frac{\mathbf{v}_1^T\mathbf{R}_{xy}\mathbf{w}_1}{\sqrt{\mathbf{v}_1^T\mathbf{R}_{xx}\mathbf{v}_1\cdot\mathbf{w}_1^T\mathbf{R}_{yy}\mathbf{w}_1}}\,. \tag{2}$$

Next, CCA identifies $\hat{\mathbf{v}}_2$ and $\hat{\mathbf{w}}_2$ such that the cross-correlation between $x_{v_2} = \hat{\mathbf{v}}_2^T\mathbf{x}$ and $y_{w_2} = \hat{\mathbf{w}}_2^T\mathbf{y}$ is maximized while x_{v_2} is uncorrelated with x_{v_1} ($E\{x_{v_1}x_{v_2}\} = 0$) and y_{w_2} is uncorrelated with y_{w_1} ($E\{y_{w_1}y_{w_2}\} = 0$), i.e.,

$$(\hat{\mathbf{v}}_2, \hat{\mathbf{w}}_2) = \underset{(\mathbf{v}_2,\mathbf{w}_2)}{\arg\max} \frac{\mathbf{v}_2^T\mathbf{R}_{xy}\mathbf{w}_2}{\sqrt{\mathbf{v}_2^T\mathbf{R}_{xx}\mathbf{v}_2\cdot\mathbf{w}_2^T\mathbf{R}_{yy}\mathbf{w}_2}} \tag{3}$$

$$\text{s.t. } \mathbf{v}_1^T\mathbf{R}_{xx}\mathbf{v}_2 = 0 \tag{4}$$

$$\mathbf{w}_1^T\mathbf{R}_{yy}\mathbf{w}_2 = 0\,. \tag{5}$$

Subsequent vectors $\hat{\mathbf{v}}_j$, $\hat{\mathbf{w}}_j$ for $j \leq \min(M,N)$ are defined similarly, with additional constraints such that $\hat{\mathbf{v}}_i^T\mathbf{R}_{xx}\hat{\mathbf{v}}_j = 0$ and $\hat{\mathbf{w}}_i^T\mathbf{R}_{yy}\hat{\mathbf{w}}_j = 0$, $\forall\, i \in \{1,\ldots,j-1\}$. The (canonical) correlation coefficients ρ_j for $j \leq \min(M,N)$ are defined as the corresponding maximized correlation coefficients, i.e.,

$$\rho_j = \frac{\hat{\mathbf{v}}_j^T\mathbf{R}_{xy}\hat{\mathbf{w}}_j}{\sqrt{\hat{\mathbf{v}}_j^T\mathbf{R}_{xx}\hat{\mathbf{v}}_j\cdot\hat{\mathbf{w}}_j^T\mathbf{R}_{yy}\hat{\mathbf{w}}_j}}\,. \tag{6}$$

Let $\hat{\mathbf{V}}$ denote the $M\times Q$ matrix that contains the first Q CCA directions $\hat{\mathbf{v}}_1,\ldots,\hat{\mathbf{v}}_Q$ in its columns and let $\hat{\mathbf{W}}$ be defined similarly for $\hat{\mathbf{w}}_1,\ldots,\hat{\mathbf{w}}_Q$. Note that $\hat{\mathbf{V}}^T\mathbf{R}_{xx}\hat{\mathbf{V}}$ and $\hat{\mathbf{W}}^T\mathbf{R}_{yy}\hat{\mathbf{W}}$ are then diagonal $Q\times Q$ matrices. We assume in the sequel that both $\mathbf{R}_{xx}$ and $\mathbf{R}_{yy}$ are invertible and well-conditioned[3]. It can then be shown that $\hat{\mathbf{V}}$ and $\hat{\mathbf{W}}$ should satisfy the eigenvalue problems [2]

$$\mathbf{R}_{xx}^{-1}\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{yx}\hat{\mathbf{V}} = \hat{\mathbf{V}}\mathbf{\Sigma}^2 \tag{7}$$

$$\mathbf{R}_{yy}^{-1}\mathbf{R}_{yx}\mathbf{R}_{xx}^{-1}\mathbf{R}_{xy}\hat{\mathbf{W}} = \hat{\mathbf{W}}\mathbf{\Sigma}^2 \tag{8}$$

where $\mathbf{\Sigma} = \text{Diag}\{\rho_1,\ldots,\rho_Q\}$. Note that the columns of $\hat{\mathbf{V}}$ and $\hat{\mathbf{W}}$ contain the eigenvectors, for which the corresponding eigenvalues are the squared canonical correlation coefficients in $\mathbf{\Sigma}^2$.

[1] When making scalability claims, we refer to the *amount* of data (in number of bits/s) that a node has to transmit to its direct neighbors, i.e., we make abstraction of possible increasing interference when the number of nodes increases.

[2] We use the term 'M-dimensional signal' to denote an M-variate stochastic process, e.g., corresponding to a multi-channel signal consisting of M channels. This means that each observation (or sample) of the signal is an M-dimensional vector.

[3] Ill-conditioned CCA problems have also been investigated [34]–[36] and can be solved using the pseudo-inverse or proper regularization, but these methods are beyond the scope of this paper.

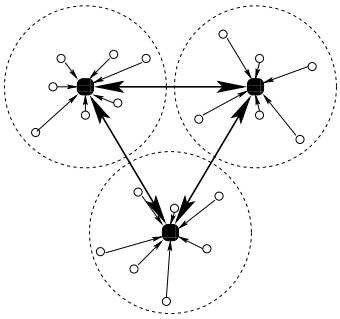

Fig. 1. A clustered WSN in which the cluster heads together form a fully-connected network.

Instead of computing the above eigenvalue decompositions (EVDs), it is seen from (7) and (8) that $\hat{\mathbf{V}}$ and $\hat{\mathbf{W}}$ can also be computed from the generalized eigenvalue decompositions (GEVDs) [37] of the symmetric matrix pencils $(\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{yx}, \mathbf{R}_{xx})$ and $(\mathbf{R}_{yx}\mathbf{R}_{xx}^{-1}\mathbf{R}_{xy}, \mathbf{R}_{yy})$, which is preferred from a numerical point of view. Furthermore, note that only one (G)EVD needs to be computed, since the solutions are related by [2]

$$\mathbf{R}_{xy}\hat{\mathbf{w}}_j = \rho_j \lambda_{j,x} \mathbf{R}_{xx}\hat{\mathbf{v}}_j \tag{9}$$

$$\mathbf{R}_{yx}\hat{\mathbf{v}}_j = \rho_j \lambda_{j,y} \mathbf{R}_{yy}\hat{\mathbf{w}}_j \tag{10}$$

where

$$\lambda_{j,x} = \lambda_{j,y}^{-1} = \sqrt{\frac{\hat{\mathbf{w}}_j^T \mathbf{R}_{yy} \hat{\mathbf{w}}_j}{\hat{\mathbf{v}}_j^T \mathbf{R}_{xx} \hat{\mathbf{v}}_j}}\,. \tag{11}$$

W.l.o.g., we will assume in the sequel that the columns of $\hat{\mathbf{V}}$ and $\hat{\mathbf{W}}$ are scaled such that $\hat{\mathbf{V}}^T\mathbf{R}_{xx}\hat{\mathbf{V}} = \hat{\mathbf{W}}^T\mathbf{R}_{yy}\hat{\mathbf{W}} = \mathbf{I}_Q$, where $\mathbf{I}_Q$ denotes the $Q \times Q$ identity matrix. Under this assumption, we find that $\lambda_{j,x} = \lambda_{j,y} = 1$.

III. Problem statement

We consider a WSN with a set of sensor nodes $\mathcal{K} = \{1, \ldots, K\}$ (we do not make any assumptions on the network topology yet). Node k collects observations of the M_k-dimensional sensor signal $\mathbf{x}_k$ and the N_k-dimensional sensor signal $\mathbf{y}_k$. Note that it is possible that either $M_k = 0$ or $N_k = 0$ at certain nodes. We assume that the stacked versions of all $\mathbf{x}_k$'s and $\mathbf{y}_k$'s yield the M-dimensional $\mathbf{x}$ and the N-dimensional $\mathbf{y}$, as defined in Subsection II, respectively, where $M = \sum_{k\in\mathcal{K}} M_k$ and $N = \sum_{k\in\mathcal{K}} N_k$.

Scenarios where $M_k > 1$ or $N_k > 1$ usually correspond to WSN architectures where the sensor nodes have multiple sensors, or where the WSN consists of a set of K cluster heads (included in $\mathcal{K}$), which collect raw sensor data from nearby sensor nodes (not included in $\mathcal{K}$). For example, Fig. 1 visualizes a clustered WSN architecture with $K = 3$ cluster heads. Cluster head k collects raw sensor observations from M_k nearby sensor nodes resulting in an M_k-channel signal $\mathbf{x}_k$. Similarly, node k also collects an N_k-channel signal $\mathbf{y}_k$ from another (or the same) set of nearby nodes.

Let $\mathbf{X}$ denote an $M \times L$ observation matrix (with $L \gg M$) containing L observations of $\mathbf{x}$ in its columns. Then ergodicity of $\mathbf{x}$ implies that $\mathbf{R}_{xx}$ can be approximated by the sample covariance matrix, i.e.,

$$\mathbf{R}_{xx} \approx \frac{1}{L}\mathbf{X}\mathbf{X}^T \tag{12}$$

and equality holds in the case of an infinite observations window, i.e., $\mathbf{R}_{xx} = \lim_{L\to\infty} \frac{1}{L}\mathbf{X}\mathbf{X}^T$. Similarly, $\mathbf{R}_{yy} \approx \frac{1}{L}\mathbf{Y}\mathbf{Y}^T$ and $\mathbf{R}_{xy} \approx \frac{1}{L}\mathbf{X}\mathbf{Y}^T$, where $\mathbf{Y}$ contains L observations of $\mathbf{y}$ in its columns. Note that node k only has access to M_k rows of $\mathbf{X}$, and N_k rows of $\mathbf{Y}$.

To estimate the Q dominant CCA directions $\hat{\mathbf{V}}$ and $\hat{\mathbf{W}}$, all nodes may transmit their observations to an FC, such that the network-wide correlation matrices $\mathbf{R}_{xx}$, $\mathbf{R}_{yy}$, and $\mathbf{R}_{xy}$ can be estimated and updated at regular time intervals (e.g., using the L most recent observations as in (12)), followed by solving the centralized CCA problem. However, transmitting the raw sensor signal observations results in a high bit rate over the wireless links, and solving a CCA problem for large M and/or N requires significant computational power at the FC.

In the DCCA algorithm presented in this paper, each node is responsible for estimating and updating a specific part of $\hat{\mathbf{V}}$ and $\hat{\mathbf{W}}$, while avoiding a centralized computation of the network-wide correlation matrices $\mathbf{R}_{xx}$, $\mathbf{R}_{yy}$, and $\mathbf{R}_{xy}$. To this end, the DCCA algorithm performs in-network data fusion. For example, in Fig. 1, the cluster heads will fuse their M_k-channel and N_k-channel signals $\mathbf{x}_k$ and $\mathbf{y}_k$, into Q-channel signals $\overline{\mathbf{x}}_k$ and $\overline{\mathbf{y}}_k$, respectively, and only exchange Q-dimensional observations of these signals with each other. This means that the amount of data that is transferred over the long-distance links will be independent of the number of nodes in each cluster.

IV. Distributed CCA in fully-connected WSNs

A. Algorithm derivation

The DCCA algorithm iteratively updates an $M \times Q$ matrix $\mathbf{V}^i$ and an $N \times Q$ matrix $\mathbf{W}^i$, where i is the iteration index, with the goal of obtaining $\lim_{i\to\infty} \mathbf{V}^i = \hat{\mathbf{V}}$ and $\lim_{i\to\infty} \mathbf{W}^i = \hat{\mathbf{W}}$. In each iteration, L new sensor signal observations will be used to update $\mathbf{V}^i$ and $\mathbf{W}^i$ into improved estimates $\mathbf{V}^{i+1}$ and $\mathbf{W}^{i+1}$. We define the partitionings

$$\mathbf{V}^i \triangleq \begin{bmatrix} \mathbf{V}_1^i \\ \vdots \\ \mathbf{V}_K^i \end{bmatrix} \tag{13}$$

$$\mathbf{W}^i \triangleq \begin{bmatrix} \mathbf{W}_1^i \\ \vdots \\ \mathbf{W}_K^i \end{bmatrix} \tag{14}$$

where $\mathbf{V}_k^i$ is the part of $\mathbf{V}^i$ that corresponds to node k (i.e., to $\mathbf{x}_k$), such that $\mathbf{V}^{i\,T}\mathbf{x} = \sum_{k\in\mathcal{K}} \mathbf{V}_k^{i\,T}\mathbf{x}_k$ (and similarly for $\mathbf{W}_k^i$ and $\mathbf{y}$). Based on these partitionings, node k is responsible for updating the submatrices $\mathbf{V}_k^i$ and $\mathbf{W}_k^i$.

To derive the distributed algorithm to update $\mathbf{V}^i$ and $\mathbf{W}^i$, we start with the observation that the CCA problem can be posed as a constrained optimization problem, i.e., it can be shown that $\hat{\mathbf{V}}$ and $\hat{\mathbf{W}}$ are a solution of [3]

$$\max_{(\mathbf{V},\mathbf{W})} f(\mathbf{V}, \mathbf{W}) \tag{15}$$

$$\text{s.t.} \;\cdot\; \mathbf{V}^T\mathbf{R}_{xx}\mathbf{V} = \mathbf{I}_Q \tag{16}$$

$$\cdot\; \mathbf{W}^T\mathbf{R}_{yy}\mathbf{W} = \mathbf{I}_Q \tag{17}$$

with

$$f\left(\mathbf{V},\mathbf{W}\right) \triangleq \mathrm{Tr}\left\{\mathbf{V}^T\mathbf{R}_{xy}\mathbf{W}\right\} \tag{18}$$

where $\mathrm{Tr}\left\{\cdot\right\}$ denotes the trace operator. The DCCA algorithm is then based on an alternating optimization (AO) procedure that iteratively solves (15)-(17), by applying some additional constraints in each iteration (we will motivate this afterwards):

1) Set $i \leftarrow 0$, $q \leftarrow 1$
2) Initialize $\mathbf{V}^0$ and $\mathbf{W}^0$ as a random $M \times Q$ matrix and a random $N \times Q$ matrix, respectively.
3) Choose $\left(\mathbf{V}^{i+1},\mathbf{W}^{i+1}\right)$ as a solution of:

$$\max_{(\mathbf{V},\mathbf{W})} f\left(\mathbf{V},\mathbf{W}\right) \tag{19}$$

$$\text{s.t. } \cdot\ \mathbf{V}^T\mathbf{R}_{xx}\mathbf{V} = \mathbf{I}_Q \tag{20}$$

$$\cdot\ \mathbf{W}^T\mathbf{R}_{yy}\mathbf{W} = \mathbf{I}_Q \tag{21}$$

$$\cdot\ \forall\, k \in \mathcal{K}\backslash\{q\}:$$

$$\mathrm{Range}\{\mathbf{V}_k\} = \mathrm{Range}\{\mathbf{V}_k^i\} \tag{22}$$

$$\mathrm{Range}\{\mathbf{W}_k\} = \mathrm{Range}\{\mathbf{W}_k^i\} \tag{23}$$

where $\mathbf{V}_k$ and $\mathbf{W}_k$ are the k-th submatrices of $\mathbf{V}$ and $\mathbf{W}$, respectively (similar to (13)-(14)), and where $\mathrm{Range}\{\mathbf{T}\}$ denotes the subspace spanned by the columns of $\mathbf{T}$.

4) $i \leftarrow i+1$ and $q \leftarrow (q \mod K)+1$.
5) Return to step 2.

It is noted that this iterative AO procedure increases $f\left(\mathbf{V},\mathbf{W}\right)$ in a monotonic fashion. Indeed, in each iteration, the AO procedure can update one particular submatrix in $\mathbf{V}$ and $\mathbf{W}$ freely (i.e., $\mathbf{V}_q$ and $\mathbf{W}_q$), while constraining the other submatrix updates such that the current column space is preserved for each submatrix. Note that this is similar to a block coordinate ascent method, in which a different subset of the optimization variables are fixed in each iteration, resulting in optimization problems with a smaller number of optimization variables. In this case, however, none of the optimization variables are fixed, but many of them are constrained to a certain subspace instead.

Despite the fact that this AO procedure is a centralized procedure requiring the network-wide matrices $\mathbf{R}_{xx}$, $\mathbf{R}_{yy}$, and $\mathbf{R}_{xy}$, the particular form of (22)-(23) allows to execute it in a distributed fashion, as explained next.

First, it is noted that the constraints (22)-(23) are equivalent to

$$\forall\, k \in \mathcal{K}\backslash\{q\},\ \exists\, \mathbf{G}_k \in \mathbb{R}^{Q\times Q} : \mathbf{V}_k = \mathbf{V}_k^i\mathbf{G}_k \tag{24}$$

$$\forall\, k \in \mathcal{K}\backslash\{q\},\ \exists\, \mathbf{H}_k \in \mathbb{R}^{Q\times Q} : \mathbf{W}_k = \mathbf{W}_k^i\mathbf{H}_k\ . \tag{25}$$

This allows us to parameterize the optimization variables $\mathbf{V}$ and $\mathbf{W}$ as

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1^i\mathbf{G}_1 \\ \vdots \\ \mathbf{V}_{q-1}^i\mathbf{G}_{q-1} \\ \mathbf{V}_q \\ \mathbf{V}_{q+1}^i\mathbf{G}_{q+1} \\ \vdots \\ \mathbf{V}_K^i\mathbf{G}_K \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \mathbf{W}_1^i\mathbf{H}_1 \\ \vdots \\ \mathbf{W}_{q-1}^i\mathbf{H}_{q-1} \\ \mathbf{W}_q \\ \mathbf{W}_{q+1}^i\mathbf{H}_{q+1} \\ \vdots \\ \mathbf{W}_K^i\mathbf{H}_K \end{bmatrix}. \tag{26}$$

To write (26) more compactly, we define the matrix

$$\mathcal{V}_k^i \triangleq \left[\begin{array}{c|c|c} \mathbf{O} & \underline{\mathcal{V}}_k^i & \mathbf{O} \\ \hline \mathbf{I}_{M_k} & \mathbf{O} & \mathbf{O} \\ \hline \mathbf{O} & \mathbf{O} & \overline{\mathcal{V}}_k^i \end{array}\right] \tag{27}$$

with $\mathbf{O}$ denoting an all-zero matrix of appropriate dimension, and where

$$\underline{\mathcal{V}}_k^i \triangleq \mathrm{Blkdiag}\left(\mathbf{V}_1^i,\ldots,\mathbf{V}_{k-1}^i\right) \tag{28}$$

$$\overline{\mathcal{V}}_k^i \triangleq \mathrm{Blkdiag}\left(\mathbf{V}_{k+1}^i,\ldots,\mathbf{V}_K^i\right) \tag{29}$$

with $\mathrm{Blkdiag}(\cdot)$ denoting the operator that generates a block-diagonal matrix with the matrices in its argument as the diagonal blocks[4]. Similarly, we define

$$\mathcal{W}_k^i \triangleq \left[\begin{array}{c|c|c} \mathbf{O} & \underline{\mathcal{W}}_k^i & \mathbf{O} \\ \hline \mathbf{I}_{M_k} & \mathbf{O} & \mathbf{O} \\ \hline \mathbf{O} & \mathbf{O} & \overline{\mathcal{W}}_k^i \end{array}\right] \tag{30}$$

where

$$\underline{\mathcal{W}}_k^i \triangleq \mathrm{Blkdiag}\left(\mathbf{W}_1^i,\ldots,\mathbf{W}_{k-1}^i\right) \tag{31}$$

$$\overline{\mathcal{W}}_k^i \triangleq \mathrm{Blkdiag}\left(\mathbf{W}_{k+1}^i,\ldots,\mathbf{W}_K^i\right)\ . \tag{32}$$

The parameterization (26) can then be written compactly as

$$\mathbf{V} = \mathcal{V}_q^i\widetilde{\mathbf{V}}_q \tag{33}$$

$$\mathbf{W} = \mathcal{W}_q^i\widetilde{\mathbf{W}}_q \tag{34}$$

where

$$\widetilde{\mathbf{V}}_q \triangleq [\mathbf{V}_q^T|\,\mathbf{G}_1^T|\,\ldots|\,\mathbf{G}_{q-1}^T|\,\mathbf{G}_{q+1}^T|\,\ldots|\,\mathbf{G}_K^T]^T \tag{35}$$

$$\widetilde{\mathbf{W}}_q \triangleq [\mathbf{W}_q^T|\,\mathbf{H}_1^T|\,\ldots|\,\mathbf{H}_{q-1}^T|\,\mathbf{H}_{q+1}^T|\,\ldots|\,\mathbf{H}_K^T]^T \tag{36}$$

define the new optimization variables. Since this parameterization allows to eliminate the constraints (22)-(23), it is found that solving (19)-(23) is then equivalent to selecting $\left(\widetilde{\mathbf{V}}_q^{i+1},\widetilde{\mathbf{W}}_q^{i+1}\right)$ as a solution of

$$\max_{(\widetilde{\mathbf{V}}_q,\widetilde{\mathbf{W}}_q)} \mathrm{Tr}\left\{\widetilde{\mathbf{V}}_q^T{\mathcal{V}_q^i}^T\mathbf{R}_{xy}\mathcal{W}_q^i\widetilde{\mathbf{W}}_q\right\} \tag{37}$$

$$\text{s.t. } \cdot\ \widetilde{\mathbf{V}}_q^T{\mathcal{V}_q^i}^T\mathbf{R}_{xx}\mathcal{V}_q^i\widetilde{\mathbf{V}}_q = \mathbf{I}_Q \tag{38}$$

$$\cdot\ \widetilde{\mathbf{W}}_q^T{\mathcal{W}_q^i}^T\mathbf{R}_{yy}\mathcal{W}_q^i\widetilde{\mathbf{W}}_q = \mathbf{I}_Q \tag{39}$$

and setting $\mathbf{V}^{i+1} = \mathcal{V}_q^i\widetilde{\mathbf{V}}_q^{i+1}$ and $\mathbf{W}^{i+1} = \mathcal{W}_q^i\widetilde{\mathbf{W}}_q^{i+1}$.

It is noted that, if $\mathcal{V}_k^i$ and $\mathcal{W}_k^i$ would be used as compression matrices on the signals $\mathbf{x}$ and $\mathbf{y}$, respectively, then we would obtain the compressed signals

$$\widetilde{\mathbf{x}}_k^i \triangleq {\mathcal{V}_k^i}^T\mathbf{x} \tag{40}$$

$$\widetilde{\mathbf{y}}_k^i \triangleq {\mathcal{W}_k^i}^T\mathbf{y} \tag{41}$$

with corresponding (cross)-correlation matrices

$$\mathbf{R}_{\tilde{x}_k\tilde{y}_k}^i = E\{\widetilde{\mathbf{x}}_k^i\widetilde{\mathbf{y}}_k^i\} = {\mathcal{V}_k^i}^T\mathbf{R}_{xy}\mathcal{W}_k^i \tag{42}$$

$$\mathbf{R}_{\tilde{x}_k\tilde{x}_k}^i = E\{\widetilde{\mathbf{x}}_k^i\widetilde{\mathbf{x}}_k^i\} = {\mathcal{V}_k^i}^T\mathbf{R}_{xx}\mathcal{V}_k^i \tag{43}$$

$$\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^i = E\{\widetilde{\mathbf{y}}_k^i\widetilde{\mathbf{y}}_k^i\} = {\mathcal{W}_k^i}^T\mathbf{R}_{yy}\mathcal{W}_k^i\ . \tag{44}$$

[4]It is noted that this is with a slight abuse of notation, since the arguments are not square matrices.

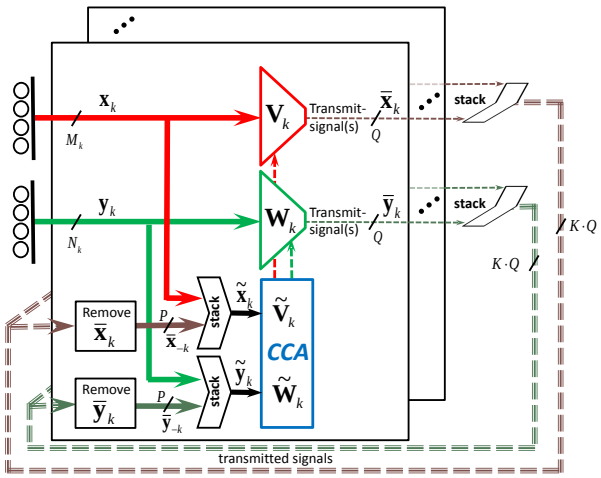

Fig. 2. A block diagram of the DCCA algorithm in a fully-connected network.

This allows us to rewrite (37)-(39) as

$$\max_{(\widetilde{\mathbf{V}}_q,\widetilde{\mathbf{W}}_q)} \operatorname{Tr}\left\{\widetilde{\mathbf{V}}_q^T \mathbf{R}^i_{\tilde{x}_q\tilde{y}_q}\widetilde{\mathbf{W}}_q\right\} \tag{45}$$

$$\text{s.t.} \cdot \ \widetilde{\mathbf{V}}_q^T \mathbf{R}^i_{\tilde{x}_q\tilde{x}_q}\widetilde{\mathbf{V}}_q = \mathbf{I}_Q \tag{46}$$

$$\cdot \ \widetilde{\mathbf{W}}_q^T \mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}\widetilde{\mathbf{W}}_q = \mathbf{I}_Q \tag{47}$$

Note that this optimization problem is again in the form of (15)-(17), and therefore corresponds to a smaller CCA problem based on the (compressed) signals $\tilde{\mathbf{x}}_q^i$ and $\tilde{\mathbf{y}}_q^i$.

The DCCA algorithm will exploit the compression of $\mathbf{x}$ and $\mathbf{y}$ based on (40)-(41), by letting each node $k \in \mathcal{K}$ compress its local sensor signals $\mathbf{x}_k$ and $\mathbf{y}_k$ into the Q-channel signals

$$\overline{\mathbf{x}}_k^i \triangleq \mathbf{V}_k^{i\,T}\mathbf{x}_k \tag{48}$$

$$\overline{\mathbf{y}}_k^i \triangleq \mathbf{W}_k^{i\,T}\mathbf{y}_k \ . \tag{49}$$

This is illustrated in Fig. 2, which shows a block diagram of the DCCA algorithm in a fully-connected network. In between iteration i and iteration $i+1$, each node compresses its L most recent observations of $\mathbf{x}_k$ and $\mathbf{y}_k$ into $\overline{\mathbf{x}}_k^i$ and $\overline{\mathbf{y}}_k^i$ and broadcasts these compressed observations to the other nodes. For the sake of an easy exposition, we assume that $Q < M_k$ and $Q < N_k$, $\forall\, k \in \mathcal{K}$. However, note that this is not a strict assumption, i.e., it is merely made for the sake of an easy notation/description of the DCCA algorithm. Indeed, if there exists a k for which $Q \geq M_k$, node k can merely broadcast uncompressed observations of $\mathbf{x}_k$ (and similarly for $\mathbf{y}_k$ when $Q \geq N_k$). Another node can then treat these raw sensor observations as part of its own sensor observations, i.e., the signals from two different nodes can be merged and together treated as the sensor signals of a single node.

Since the network is assumed to be fully connected, each node k collects observations of

$$\widetilde{\mathbf{x}}_k^i = \begin{bmatrix} \mathbf{x}_k \\ \overline{\mathbf{x}}_{-k}^i \end{bmatrix} \tag{58}$$

$$\widetilde{\mathbf{y}}_k^i = \begin{bmatrix} \mathbf{y}_k \\ \overline{\mathbf{y}}_{-k}^i \end{bmatrix} \tag{59}$$

TABLE I
THE DCCA ALGORITHM IN A FULLY-CONNECTED WSN

1) Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all $\mathbf{V}_k^0$ and $\mathbf{W}_k^0$, $\forall\, k \in \mathcal{K}$, with random entries.
2) Each node $k \in \mathcal{K}$ broadcasts L compressed observations $\overline{\mathbf{x}}_k^i[j] = \mathbf{V}_k^{i\,T}\mathbf{x}_k[iL+j]$ and $\overline{\mathbf{y}}_k^i[j] = \mathbf{W}_k^{i\,T}\mathbf{y}_k[iL+j]$ (where $j = 1 \ldots L$).
3) At node q:
   - Estimate $\mathbf{R}^i_{\tilde{x}_q\tilde{x}_q}$, $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$ and $\mathbf{R}^i_{\tilde{x}_q\tilde{y}_q}$ based on the L new observations of the signals $\widetilde{\mathbf{x}}_q^i$ and $\widetilde{\mathbf{y}}_q^i$ as defined in (58)-(59).
   - Compute the columns of $\widetilde{\mathbf{V}}_q^{i+1}$ and $\widetilde{\mathbf{W}}_q^{i+1}$ as the Q principal CCA directions between $\widetilde{\mathbf{x}}_q^i$ and $\widetilde{\mathbf{y}}_q^i$.
   - Define $P = Q(K-1)$ and partition $\widetilde{\mathbf{V}}_q^{i+1}$ and $\widetilde{\mathbf{W}}_q^{i+1}$ as

$$\mathbf{V}_q^{i+1} = \left[\mathbf{I}_{M_q}\ \mathbf{O}_{M_q\times P}\right]\widetilde{\mathbf{V}}_q^{i+1} \tag{50}$$

$$\mathbf{G}_{-q} = \left[\mathbf{O}_{P\times M_q}\ \mathbf{I}_P\right]\widetilde{\mathbf{V}}_q^{i+1} \tag{51}$$

$$\mathbf{W}_q^{i+1} = \left[\mathbf{I}_{N_q}\ \mathbf{O}_{N_q\times P}\right]\widetilde{\mathbf{W}}_q^{i+1} \tag{52}$$

$$\mathbf{H}_{-q} = \left[\mathbf{O}_{P\times N_q}\ \mathbf{I}_P\right]\widetilde{\mathbf{W}}_q^{i+1} \tag{53}$$

   and broadcast $\mathbf{G}_{-q}$ and $\mathbf{H}_{-q}$ to all other nodes.
4) Each node $k \in \mathcal{K}\backslash\{q\}$ updates

$$\mathbf{V}_k^{i+1} = \mathbf{V}_k^i\mathbf{G}_k \tag{54}$$

$$\mathbf{W}_k^{i+1} = \mathbf{W}_k^i\mathbf{H}_k \tag{55}$$

   where

$$\mathbf{G}_{-q} = \left[\mathbf{G}_1^T \ \ldots\ \mathbf{G}_{q-1}^T\,\mathbf{G}_{q+1}^T \ \ldots\ \mathbf{G}_K^T\right]^T \tag{56}$$

$$\mathbf{H}_{-q} = \left[\mathbf{H}_1^T \ \ldots\ \mathbf{H}_{q-1}^T\,\mathbf{H}_{q+1}^T \ \ldots\ \mathbf{H}_K^T\right]^T . \tag{57}$$

5) $i \leftarrow i+1$ and $q \leftarrow (q \mod K)+1$.
6) Return to step 2.

where $\overline{\mathbf{x}}_{-k}^i \triangleq [\overline{\mathbf{x}}_1^{i\,T} \ \ldots\ \overline{\mathbf{x}}_{k-1}^{i\,T}\,\overline{\mathbf{x}}_{k+1}^{i\,T} \ \ldots\ \overline{\mathbf{x}}_K^{i\,T}]^T$ and $\overline{\mathbf{y}}_{-k}^i \triangleq [\overline{\mathbf{y}}_1^{i\,T} \ \ldots\ \overline{\mathbf{y}}_{k-1}^{i\,T}\,\overline{\mathbf{y}}_{k+1}^{i\,T} \ \ldots\ \overline{\mathbf{y}}_K^{i\,T}]^T$. Note that (58)-(59) is indeed consistent with the earlier definition (40)-(41).

Similar to (12), $\mathbf{R}^i_{\tilde{x}_q\tilde{x}_q}$, $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$, and $\mathbf{R}^i_{\tilde{x}_q\tilde{y}_q}$ can be estimated at node q as it has access to observations of $\widetilde{\mathbf{x}}_q^i$ and $\widetilde{\mathbf{y}}_q^i$, hence the corresponding smaller CCA problem (45)-(47) can indeed be solved locally. The DCCA algorithm exactly performs these operations, and is described in detail in Table I. In each iteration of the DCCA algorithm, one particular node q solves a local CCA problem defined by (45)-(47), thereby freely choosing its local parameters $\mathbf{V}_q$ and $\mathbf{W}_q$ and effectively transforming the columns of the $\mathbf{V}_k$'s and $\mathbf{W}_k$'s at the other nodes $k \in \mathcal{K}\backslash\{q\}$ with $Q \times Q$ transformation matrices.

As the DCCA algorithm corresponds to the original AO procedure, and since the latter has a monotonic increase of $f(\mathbf{V},\mathbf{W})$ under the constraints (16)-(17), we readily obtain the following result:

Result IV.1. *The objective function $f\left(\mathbf{V}^i,\mathbf{W}^i\right)$ increases monotonically in each iteration of the DCCA algorithm, i.e., $f\left(\mathbf{V}^{i+1},\mathbf{W}^{i+1}\right) \geq f\left(\mathbf{V}^i,\mathbf{W}^i\right)$, $\forall\, i \in \mathbb{N}_0$. Furthermore, all matrices $\mathbf{V}^i$ and $\mathbf{W}^i$, $\forall i \in \mathbb{N}_0$, satisfy $\mathbf{V}^{i\,T}\mathbf{R}_{xx}\mathbf{V}^i = \mathbf{I}_Q$ and $\mathbf{W}^{i\,T}\mathbf{R}_{yy}\mathbf{W}^i = \mathbf{I}_Q$, respectively.*

It is noted that the above statement does not necessarily hold for $i = 0$ because the DCCA algorithm can be initialized with random matrices, which are possibly not normalized, i.e, $\mathbf{V}^{0\,T}\mathbf{R}_{xx}\mathbf{V}^0 \neq \mathbf{I}_Q$ and/or $\mathbf{W}^{0\,T}\mathbf{R}_{yy}\mathbf{W}^0 \neq \mathbf{I}_Q$.

Untuitively, since $\hat{\mathbf{V}}$ and $\hat{\mathbf{W}}$ maximize $f(\mathbf{V},\mathbf{W})$ under (16)-(17), it is expected that the DCCA algorithm converges to the optimal CCA solution due to the monotonic increase of $f\left(\mathbf{V}^i,\mathbf{W}^i\right)$. A formal convergence proof is given in the next section. It is noted that, if $(\mathbf{V}^i,\mathbf{W}^i) \rightarrow (\hat{\mathbf{V}},\hat{\mathbf{W}})$, then also the Q largest canonical correlation coefficients of the local CCA problems at the different nodes converge to the Q largest canonical correlation coefficients $(\rho_1,\ldots,\rho_Q)$ of the network-wide CCA problem.

Remark IV.1. The DCCA algorithm is assumed to operate in an adaptive (time-recursive) context, and therefore all nodes collect and broadcast different sensor signal observations in each iteration. The number of observations L that are collected and transmitted in between the iterations (step 2) should allow for a sufficiently accurate estimate of $\mathbf{R}^i_{\tilde{x}_q\tilde{x}_q}$, $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$, and $\mathbf{R}^i_{\tilde{x}_q\tilde{y}_q}$ in step 3. The transmission of $\mathbf{G}_{-q}$ and $\mathbf{H}_{-q}$ is therefore negligible compared to the transmission of L observations of $\overline{\mathbf{x}}^i_k$ and $\overline{\mathbf{y}}^i_k$ in each iteration. In principle, the transmission of $\mathbf{G}_{-q}$ and $\mathbf{H}_{-q}$ and the updates (54)-(55) can even be omitted, as they do not have any influence on the convergence of the algorithm (the convergence proof in Subsection IV-B can easily be modified to this case). However, note that this introduces a sign ambiguity within the different subblocks of $\mathbf{V}^i$, i.e., it can then be shown that

$$i \rightarrow \infty: \ \mathbf{G}_{-q} = [\pm\mathbf{I}_Q \ \ldots \ \pm\mathbf{I}_Q]^T \tag{60}$$

(and similarly for $\mathbf{H}_{-q}$). Furthermore, the concatenation of the $\mathbf{V}^i_k$'s and the $\mathbf{W}^i_k$'s will not satisfy the constraints (20)-(21) anymore, until the algorithm has converged.

Remark IV.2. Assuming that the different iterations of the DCCA algorithm are spread over different sample blocks, the DCCA algorithm provides a significant reduction in communication cost at each node $k \in \mathcal{K}$ (assuming $2Q \ll M_k + N_k$), i.e., node k broadcasts observations of two Q-dimensional signals ($\overline{\mathbf{x}}_k$ and $\overline{\mathbf{y}}_k$) instead of the observations of M_k- and N_k-dimensional signals ($\mathbf{x}_k$ and $\mathbf{y}_k$). This results in a communication load of $O(2Q)$ per node. Furthermore, a simple scaling argument shows that the DCCA algorithm also has a reduced computational complexity compared to a centralized CCA algorithm. This is due to the reduced dimension of the local CCA problem that is solved in each iteration. However, this comes at a cost of a slower tracking performance when applied in an adaptive context, due to the iterative nature of the algorithm.

Remark IV.3. It is noted that we have implicitely made two pragmatic assumptions[5] to guarantee that $\widetilde{\mathbf{V}}^{i+1}_q$ and $\widetilde{\mathbf{W}}^{i+1}_q$ are well-defined (up to a sign ambiguity) in each iteration of the DCCA algorithm:

- The matrices $\mathbf{R}^i_{\tilde{x}_q\tilde{x}_q}$ and $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$ have full rank, $\forall\, i \in \mathbb{N}$.
- The $Q+1$ largest canonical correlation coefficients of $\tilde{\mathbf{x}}^i_q$ and $\tilde{\mathbf{y}}^i_q$ are unique in each iteration,i.e.,

$$\exists \epsilon > 0, \forall i \in \mathbb{N}, \forall n \in \{1,\ldots,Q\} : \tilde{\rho}^i_n - \tilde{\rho}^i_{n+1} > \epsilon \tag{61}$$

where $\tilde{\rho}^i_n$ denotes the n-th largest canonical correlation coefficient between $\widetilde{\mathbf{x}}^i_q$ and $\widetilde{\mathbf{y}}^i_q$, and where node q is the updating node in iteration i.

It is noted that these assumptions are merely made for the sake of mathematical tractability. Although they are mostly satisfied in practice, we briefly describe in Appendix C how the DCCA algorithm can be modified if these assumptions are violated. These modifications will also resolve the sign ambiguity in the columns of $\mathbf{V}^{i+1}$ and $\mathbf{W}^{i+1}$ (see also Subsection IV-B).

Remark IV.4. An additional trade-off between computational complexity and convergence speed can be achieved by reducing the dimensions of $\widetilde{\mathbf{x}}^i_q$ and $\widetilde{\mathbf{y}}^i_q$ at node q. This can be done by, e.g., summing the received observations, rather then incorporating the received observations of each $(\overline{\mathbf{x}}^i_k, \overline{\mathbf{y}}^i_k)$, $\forall\, k \in \mathcal{K}\backslash\{q\}$, separately. In this case, (58) is redefined as

$$\widetilde{\mathbf{x}}^i_k = \begin{bmatrix} \mathbf{x}_k \\ \sum_{q\in\mathcal{K}\backslash\{k\}} \overline{\mathbf{x}}^i_q \end{bmatrix} \tag{62}$$

and several other variables have to be redefined accordingly (details omitted). This indeed reduces the computational complexity at node q, but it also reduces the degrees of freedom per updating step, yielding a slower convergence and hence slower adaptation in time-varying scenarios. This is also the reason why the DCCA algorithm applied in a tree topology (see Section VI) converges slower than when it is applied in a fully-connected network topology.

B. Convergence analysis

In this subsection, we provide two results that together prove convergence and optimality of the DCCA algorithm. It is noted that the CCA solution always has a sign ambiguity, which we will pragmatically ignore to not overcomplicate the convergence statements and proofs. With a slight abuse of notation, we write $\mathbf{V}^i = \mathbf{V}^*$ and $\lim_{i\rightarrow\infty}\mathbf{V}^i = \mathbf{V}^*$ to denote equality/convergence up to a per-column sign ambiguity. It is noted that the sign ambiguity may indeed hamper convergence due to arbitrary sign changes over the different iterations. However, these can be easily avoided if in each iteration proper signs are selected such that $\|\mathbf{V}^{i+1} - \mathbf{V}^i\|_F$ is minimized (see also Appendix C).

An equilibrium point of the DCCA algorithm is defined as a point $(\mathbf{V}^*,\mathbf{W}^*)$ such that, if $\left(\mathbf{V}^i,\mathbf{W}^i\right) = (\mathbf{V}^*,\mathbf{W}^*)$ at iteration $i = i^*$, then $\left(\mathbf{V}^{j+1},\mathbf{W}^{j+1}\right) = \left(\mathbf{V}^j,\mathbf{W}^j\right)$, $\forall\, j \geq i^*$. An equilibrium point is assumed to be unstable under the DCCA algorithm update rules if a small perturbation on the equilibrium point may cause the DCCA algorithm to diverge away from the equilibrium point. The following statement addresses the equilibrium point(s) of the DCCA algorithm and their stability.

Result IV.2. *Let $\mathcal{E}$ denote the set of all equilibrium points of the DCCA algorithm and let $(\mathbf{V}^*,\mathbf{W}^*) \in \mathcal{E}$. Then the columns of $\mathbf{V}^*$ and $\mathbf{W}^*$ can only contain CCA directions for $\mathbf{x}$ and $\mathbf{y}$. Furthermore, $\mathcal{E}$ always contains $(\hat{\mathbf{V}},\hat{\mathbf{W}})$, which has*

[5]It is noted that similar assumptions would appear in a centralized CCA implementation to extract Q principal CCA directions. This is not surprising, given the fact that step 3 in the DCCA algorithm basically solves a (local) CCA problem.

the Q principal CCA directions, and this is the only stable equilibrium point under the DCCA update rules.

Proof: See Appendix A. ∎

Basically, Result IV.2 says that $(\hat{\mathbf{V}}, \hat{\mathbf{W}})$ is a stable equilibrium point of the DCCA algorithm, but that other CCA directions may also form equilibrium points. However, it is highly unlikely that such additional equilibrium points exist, since this would imply that there exists a CCA direction that is a maximum of the objective function $f(\mathbf{V}, \mathbf{W})$ over K different constraint sets defined by (20)-(23), $\forall\, q \in \mathcal{K}$. This usually only holds for the principal CCA directions $(\hat{\mathbf{V}}, \hat{\mathbf{W}})$. Moreover, even in the rare cases where $\mathcal{E}$ is not a singleton, the suboptimal equilibrium points will be unstable.

Finally, convergence of the DCCA algorithm is established:

Result IV.3. *For any initialization of the DCCA algorithm, the limit $\lim_{i\to\infty}(\mathbf{V}^i, \mathbf{W}^i)$ exists, i.e., the DCCA algorithm converges.*

Proof: See Appendix B. ∎

Due to inevitable estimation errors and numerical noise, we can safely assume that -in practice- the DCCA algorithm will diverge away from possible unstable equilibrium points (in the rare cases where these indeed exist). Since $(\hat{\mathbf{V}}, \hat{\mathbf{W}})$ is the only stable equilibrium point (see Result IV.2), we conclude from Result IV.3 that the DCCA algorithm converges to the Q principal CCA directions, i.e., $\lim_{i\to\infty}(\mathbf{V}^i, \mathbf{W}^i) = (\hat{\mathbf{V}}, \hat{\mathbf{W}})$.

V. Distributed blind source separation

In this section, we briefly demonstrate how DCCA can be used for distributed BSS in a WSN.

A. CCA-based BSS

We consider the following data model that generates observations of the M-dimensional signal $\mathbf{x}$:

$$\mathbf{x}[t] = \mathbf{A}\mathbf{s}[t] + \mathbf{n}[t] \tag{63}$$

where $\mathbf{s} = [s_1 \ldots s_J]^T$ is a J-dimensional signal containing J mutually uncorrelated source signals, $\mathbf{A}$ is an $M \times J$ mixing matrix, and $\mathbf{n}$ is a noise term. The goal of BSS is to compute a demixing matrix $\mathbf{V}$ such that $\mathbf{V}^T\mathbf{x}$ is approximately equal to the original source signals $\mathbf{s}$ up to an unknown scaling and permutation. For mathematical tractability[6], we make the following assumptions:

- We consider the noiseless case, i.e., $\mathbf{n}[t] = \mathbf{0}$, $\forall\, t \in \mathbb{N}$. This is a good approximation if $\mathbf{x}$ is observed at a high signal-to-noise ratio (SNR). If the influence of the noise is too high, then the noise covariance matrix must be known (or estimated online) such that it can be subtracted from the covariance matrices defined in the sequel. In [22], an alternative CCA-based approach is proposed for noisy BSS.
- We assume that the mixing matrix $\mathbf{A}$ is fixed, i.e., it does not change over time. However, adaptive CCA-based approaches have also been described in the literature [21]–[23]. It is noted that the proposed DCCA algorithm can also track slow changes in $\mathbf{A}$, i.e., if $\mathbf{A}$ changes slower than the actual convergence time of the DCCA algorithm.
- We assume that the source signals in $\mathbf{s}$ are stationary, i.e., their autocorrelation is independent of t.
- The sources of interest all have a different autocorrelation structure (see below).

[6]Although some of these assumptions may appear to be very restrictive, it has been demonstrated that CCA-based BSS often works well, even when some of these assumptions are not entirely satisfied [7], [10], [12].

The last assumption is the most crucial, since it is explicitely exploited by CCA-based BSS algorithms, as explained next. If it is not satisfied, other BSS methods should be used.

The basic operation is a CCA between $\mathbf{x}$ as defined in (63) and $\mathbf{y}$ which is a delayed version of $\mathbf{x}$ with time lag d, i.e.,

$$\mathbf{y}[t] \triangleq \mathbf{x}[t-d]\ . \tag{64}$$

Let

$$r_{s_i}[\tau] \triangleq \frac{E\{s_i[t]s_i[t-\tau]\}}{E\{s_i[t]^2\}} \tag{65}$$

denote the τ-lag autocorrelation of the i-th source signal in $\mathbf{s}$, then the basic requirement is that there exists a time lag d and a source index $i \in \{1, \ldots, J\}$ such that

$$\forall\, j \in \{1, \ldots, J\}\backslash\{i\} :\ r_{s_i}[d] > r_{s_j}[d]\ . \tag{66}$$

If this holds, then a CCA based on $\mathbf{x}$ and $\mathbf{y}$ can be used to extract source s_i from the mixture (63) up to an unknown scaling. This can be explained by the fact that, for any time lag d, the autocorrelation of a sum of signals is always smaller than or equal to the autocorrelation of the signal with maximum autocorrelation, i.e.,

$$r_{s_i+s_j}[\tau] \leq \max(r_{s_i}[\tau], r_{s_j}[\tau])\ . \tag{67}$$

Therefore, a CCA will provide the directions $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{w}}_1$ such that the source with highest autocorrelation at time lag d is extracted, as this will yield a higher correlation coefficient between $\hat{\mathbf{v}}_1^T\mathbf{x}$ and $\hat{\mathbf{w}}_1^T\mathbf{y}$. Indeed, (67) implies that the first canonical correlation coefficient ρ_1 is maximized if $\hat{\mathbf{v}}_1^T\mathbf{A}$ contains only one non-zero element, i.e., at the position corresponding to the source with highest autocorrelation at time lag d, i.e., s_i. It therefore holds that $s_i[t] = \hat{\mathbf{v}}_1^T\mathbf{x}[t]$.

Based on a similar reasoning, it can be shown that Q source signals $s_{i_1}, \ldots, s_{i_Q}$ can be extracted, if

$$\begin{aligned}&\forall\, j \in \{1, \ldots, J\}\backslash\{i_1, \ldots, i_Q\} :\\ &r_{s_{i_1}}[d] > r_{s_{i_2}}[d] > \ldots > r_{s_{i_Q}}[d] > r_{s_j}[d]\ .\end{aligned} \tag{68}$$

We then have that

$$\begin{bmatrix} s_{i_1}[t] \\ \vdots \\ s_{i_Q}[t] \end{bmatrix} = \hat{\mathbf{V}}^T\mathbf{x}[t] \tag{69}$$

where the columns of $\hat{\mathbf{V}}$ are the Q principal CCA directions.

In case the actual auto-correlation structure of the hidden sources is unknown, it is often useful to consider multiple time lags. In [20], it is proposed to define $\mathbf{y}$ as a stacked version

of D different time lags of $\mathbf{x}$, i.e.,

$$\mathbf{y}[t] \triangleq \left[\mathbf{x}[t-1]^T \ \ldots \ \mathbf{x}[t-D]^T\right]^T . \tag{70}$$

A CCA between the M-dimensional signal $\mathbf{x}$ and the (MD)-dimensional signal $\mathbf{y}$ is then computed.

B. DCCA-based BSS

The CCA-based BSS approach described in Subsection V-A can straightforwardly be applied in a distributed context using the proposed DCCA algorithm. Consider a WSN where node $k \in \mathcal{K}$ collects observations of M_k different sensor signals stacked in $\mathbf{x}_k$. Let $\mathbf{x}$, as defined in (63), denote the stacked version of all $\mathbf{x}_k$'s, $\forall\, k \in \mathcal{K}$. Define $\mathbf{y}_k$ as the stacked vector of D different time-delayed versions of $\mathbf{x}_k$, e.g., such that the stacked vector of all $\mathbf{y}_k$'s, $\forall\, k \in \mathcal{K}$, is equal to (70).

Applying the DCCA algorithm to extract Q sources from the sensor signals in $\mathbf{x}$ requires each node $k \in \mathcal{K}$ to broadcast observations of two Q-dimensional (compressed) signals, i.e., $\overline{\mathbf{x}}_k^i$ and $\overline{\mathbf{y}}_k^i$, as defined in (48). After convergence of the DCCA algorithm, the Q source signals are found as

$$\bar{\mathbf{s}} \triangleq \lim_{i\to\infty} \sum_{k\in\mathcal{K}} \overline{\mathbf{x}}_k^i \tag{71}$$

where $\bar{\mathbf{s}}$ contains the Q signals from $\mathbf{s}$ with the largest autocorrelation over D time lags. Since each node has access to the observations of all the $\overline{\mathbf{x}}_k^i$'s, $\forall\, k \in \mathcal{K}$, each node can reconstruct observations of the Q source signals.

Remark V.1. It is noted that the required bit rate over the wireless links is independent of the number of time lags D that are taken into account, i.e., each node $k \in \mathcal{K}$ broadcasts observations of $\overline{\mathbf{x}}_k^i$ and $\overline{\mathbf{y}}_k^i$, where the dimension of $\overline{\mathbf{y}}_k^i$ is Q, independent of the dimension of $\mathbf{y}$. In the case where $D = 1$, i.e., only a single time lag is considered, the required bit rate can even be reduced by a factor of two. This can be explained as follows. First, it is noted that the matrix $\mathbf{R}_{xy}$ becomes a symmetric matrix, i.e.,

$$\mathbf{R}_{xy} = \mathbf{A} E\{\mathbf{s}[t]\mathbf{s}[t-1]\}\mathbf{A}^T \tag{72}$$

where $E\{\mathbf{s}[t]\mathbf{s}[t-1]\}$ is a diagonal matrix. Furthermore, we find that $\mathbf{R}_{xx} = \mathbf{R}_{yy}$. In this case, (7) and (8) become equivalent, and therefore $\hat{\mathbf{V}} = \hat{\mathbf{W}}$. This also holds for the local CCA problems at the individual nodes, and therefore

$$\overline{\mathbf{y}}_k^i[t] = \overline{\mathbf{x}}_k^i[t-1] \tag{73}$$

hence the observations of $\overline{\mathbf{y}}_k^i$ do not have to be broadcast, as they are merely delayed versions of the observations of $\overline{\mathbf{x}}_k^i$.

VI. The DCCA algorithm in networks with a tree topology

In this section, we explain how the DCCA algorithm can be modified to operate in a partially-connected network, where it is assumed that the network has been pruned to a tree topology (this will be motivated in Section VI-B). Before describing the algorithm, we first explain how the data flow is organized in such networks. For the sake of an easy exposition, we first focus on the data flow in a star topology network and we then extend this result to general tree topologies.

In the sequel, we define $\mathcal{N}_k$ as the set of neighbors of node k, i.e., nodes that are connected to node k (by definition $k \notin \mathcal{N}_k$). The nodes with a single neighbor are referred to as leaf nodes.

A. Data flow in star topology networks

We first assume that the network has a star topology, where all the nodes are leaf nodes, except for a central node c for which $\mathcal{N}_c = \mathcal{K}\backslash\{c\}$. We assume that the central node c transmits (broadcasts) the same data to all the leaf nodes. We redefine the signal of which observations are broadcast by the central node c to all the leaf nodes as

$$\overline{\mathbf{x}}_c^i \triangleq \mathbf{V}_c^{i\,T}\mathbf{x}_c + \sum_{l\in\mathcal{N}_c} \overline{\mathbf{x}}_{lc}^i \tag{74}$$

$$\overline{\mathbf{y}}_c^i \triangleq \mathbf{W}_c^{i\,T}\mathbf{y}_c + \sum_{l\in\mathcal{N}_c} \overline{\mathbf{y}}_{lc}^i \tag{75}$$

where $\overline{\mathbf{x}}_{lc}^i \triangleq \mathbf{V}_l^{i\,T}\mathbf{x}_l$ and $\overline{\mathbf{y}}_{lc}^i \triangleq \mathbf{W}_l^{i\,T}\mathbf{y}_l$ denotes the signals of which observations are transmitted from a leaf node $l \in \mathcal{N}_c$ to the central node c. This definition also implies a causal relationship in the data exchange between the nodes, i.e., the leaf nodes first transmit observations of their respective $\overline{\mathbf{x}}_{lc}^i$'s and $\overline{\mathbf{y}}_{lc}^i$'s ($l \neq c$) to node c, after which node c computes the corresponding observations of $\overline{\mathbf{x}}_c^i$ and $\overline{\mathbf{y}}_c^i$ and broadcasts these to the leaf nodes.

It is important to note that the observations of $\overline{\mathbf{x}}_c^i$ that are received at a leaf node l also contain a contribution from node l's own observations of $\overline{\mathbf{x}}_{lc}^i$ (See (74)). This introduces a feedback path which affects the algorithm dynamics, as well as the equilibrium set (see also [32], [33]). This feedback phenomenon in fact eliminates the monotonic increase of $f\left(\mathbf{V}^i, \mathbf{W}^i\right)$ and results in convergence problems.

Two different solutions have been described in [32], [33] to tackle a similar feedback problem. The first solution is referred to as transmitter feedback cancellation (TFC) and matches well with point-to-point communication protocols, in which each connected node pair has a reserved communication link. In this case, the center node c can send different data to each of the leaf nodes. Let $\overline{\mathbf{x}}_{ck}^i$ denote the signal of which node c sends observations to node k, then the feedback path is removed by setting (compare with (74))

$$\overline{\mathbf{x}}_{ck}^i \triangleq \mathbf{V}_c^{i\,T}\mathbf{x}_c + \sum_{l\in\mathcal{N}_c\backslash\{k\}} \overline{\mathbf{x}}_{lc}^i . \tag{76}$$

The second solution is referred to as receiver feedback cancellation (RFC) [33], where the central node still broadcasts the same data to all the leaf nodes. In this case, the leaf nodes remove their own contribution from the observations of $\overline{\mathbf{x}}_c^i$, i.e., node k effectively uses the feedback-corrected input signal

$$\overline{\mathbf{x}}_{c,-k}^i \triangleq \overline{\mathbf{x}}_c^i - \overline{\mathbf{x}}_{kc}^i . \tag{77}$$

The RFC approach is clearly much more efficient in terms of communication, because the signal $\overline{\mathbf{x}}_c^i$ can be broadcast to all the leaf nodes at once.

It is noted that $\overline{\mathbf{x}}^i_{c,-k} = \overline{\mathbf{x}}^i_{ck}$, i.e., the DCCA algorithm behaves exactly the same, whether we apply TFC or RFC. Therefore, we do not have to distinguish between both cases in the sequel, i.e., we will only focus on TFC for the sake of an easier exposition. Finally, it is noted that a similar TFC or RFC approach is applied to define $\overline{\mathbf{y}}^i_{ck}$ and $\overline{\mathbf{y}}^i_{c,-k}$.

B. Data flow in tree topology networks

Let $\overline{\mathbf{x}}^i_{kq}$ and $\overline{\mathbf{y}}^i_{kq}$ denote the signals of which observations are transmitted from node k to node q. The definition of the $\overline{\mathbf{x}}^i_{kq}$ and $\overline{\mathbf{y}}^i_{kq}$ signals within a tree topology, as well as the corresponding data flow, are similarly defined as in (76), here repeated for convenience:

$$\overline{\mathbf{x}}^i_{kq} \triangleq \mathbf{V}^{i\,T}_k \mathbf{x}_k + \sum_{l\in\mathcal{N}_k\backslash\{q\}} \overline{\mathbf{x}}^i_{lk} \tag{78}$$

$$\overline{\mathbf{y}}^i_{kq} \triangleq \mathbf{W}^{i\,T}_k \mathbf{y}_k + \sum_{l\in\mathcal{N}_k\backslash\{q\}} \overline{\mathbf{y}}^i_{lk} \,. \tag{79}$$

It is noted that, due to the fusion of the local sensor signals ($\mathbf{x}_k$ and $\mathbf{y}_k$) with the signals that are transmitted by the neighboring nodes ($\overline{\mathbf{x}}^i_{lk}$ and $\overline{\mathbf{y}}^i_{lk}$), a significant compression is obtained[7], even if Q is larger than the local sensor signal dimensions, i.e., even if $Q > M_k$ and/or $Q > N_k$.

The motivation for pruning the network to a tree topology, is based on the observations in Section VI-A, and can be summarized as follows:

1) Tree topologies have no cycles, and therefore all feedback paths are eliminated (if RFC or TFC is applied).
2) In a tree topology, there is a natural order to compute the observations of the different $\overline{\mathbf{x}}^i_{kq}$'s and $\overline{\mathbf{y}}^i_{kq}$'s as defined in (78)-(79), i.e., all the dependencies are easily resolved (see also below). This is not the case in network graphs with loops, where deadlock situations may arise.

In a tree topology, the computation of the $\overline{\mathbf{x}}^i_{kq}$'s and the corresponding data flow can be completely data-driven without any central coordination: a node k computes and transmits L observations of $\overline{\mathbf{x}}^i_{kq}$ as soon as it has received L observations of $\overline{\mathbf{x}}^i_{lk}$ from its neighbors $l \in \mathcal{N}_k\backslash\{q\}$. Since any leaf node k only has one neighbor (say, q), $\overline{\mathbf{x}}^i_{kq} = \mathbf{V}^{i\,T}_k \mathbf{x}_k$, which does not rely on observations from any other node, and therefore the leaf nodes will initiate the computation of (78). This data-driven approach will naturally result in a so-called 'fusion flow' from the leaf nodes to the root node(s), followed by a so-called 'diffusion flow' from the root node(s) to the leaf nodes (see [33]).

Again, it is noted that a similar reasoning can be followed for the compressed signals corresponding to the $\mathbf{y}_k$'s.

C. The DCCA algorithm in tree topology networks

The vector $\widetilde{\mathbf{x}}^i_k$, which contains all the signals of which observations are available to node k, is redefined as (compare with (58))

$$\widetilde{\mathbf{x}}^i_k \triangleq \begin{bmatrix} \mathbf{x}_k \\ \overline{\mathbf{x}}^i_{\rightarrow k} \end{bmatrix} \tag{86}$$

[7]This should be compared to the relay case, i.e., where all the raw sensor observations are individually relayed throughout the network.

TABLE II
THE DCCA ALGORITHM IN A WSN WITH A TREE TOPOLOGY.

1) Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all $\mathbf{V}^0_k$ and $\mathbf{W}^0_k$, $\forall\, k \in \mathcal{K}$, with random entries.
2) Each node $k \in \mathcal{K}$ transmits L observations of the fused signals $\overline{\mathbf{x}}^i_{kl}$ and $\overline{\mathbf{y}}^i_{kl}$ to node l, $\forall\, l \in \mathcal{N}_k$, based on (78)-(79). The order in which these are computed and forwarded by the different nodes is dictated by (78)-(79) (which initiates at the leaf nodes).
3) At node q:
   - Estimate $\mathbf{R}^i_{\tilde{x}_q\tilde{x}_q}$, $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$ and $\mathbf{R}^i_{\tilde{x}_q\tilde{y}_q}$ based on the L new observations of the signals $\widetilde{\mathbf{x}}^i_q$ and $\widetilde{\mathbf{y}}^i_q$ as defined in (86)-(87).
   - Compute the columns of $\widetilde{\mathbf{V}}^{i+1}_q$ and $\widetilde{\mathbf{W}}^{i+1}_q$ as the Q principal CCA directions between $\widetilde{\mathbf{x}}^i_q$ and $\widetilde{\mathbf{y}}^i_q$.
   - Define $P = Q|\mathcal{N}_q|$ (with $|\cdot|$ denoting cardinality) and partition $\widetilde{\mathbf{V}}^{i+1}_q$ and $\widetilde{\mathbf{W}}^{i+1}_q$ as

$$\mathbf{V}^{i+1}_q = \left[\mathbf{I}_{M_q}\ \mathbf{O}_{M_q\times P}\right]\widetilde{\mathbf{V}}^{i+1}_q \tag{80}$$

$$\mathbf{G}_{\rightarrow q} = \left[\mathbf{O}_{P\times M_q}\ \mathbf{I}_P\right]\widetilde{\mathbf{V}}^{i+1}_q \tag{81}$$

$$\mathbf{W}^{i+1}_q = \left[\mathbf{I}_{N_q}\ \mathbf{O}_{N_q\times P}\right]\widetilde{\mathbf{W}}^{i+1}_q \tag{82}$$

$$\mathbf{H}_{\rightarrow q} = \left[\mathbf{O}_{P\times N_q}\ \mathbf{I}_P\right]\widetilde{\mathbf{W}}^{i+1}_q \,. \tag{83}$$

4) Define the partitioning $\mathbf{G}_{\rightarrow q} = \left[\mathbf{G}^T_{l_1}\ \ldots\ \mathbf{G}^T_{l_{n_q}}\right]^T$ and $\mathbf{H}_{\rightarrow q} = \left[\mathbf{H}^T_{l_1}\ \ldots\ \mathbf{H}^T_{l_{n_q}}\right]^T$ where each $\mathbf{G}_{l_k}$ and $\mathbf{H}_{l_k}$ is a $Q\times Q$ matrix and where $\{l_1,\ldots,l_{n_q}\} = \mathcal{N}_q$. Disseminate $\mathbf{G}_l$ and $\mathbf{H}_l$ over the tree branch $\mathcal{B}_{lq}$, $\forall\, l \in \mathcal{N}_q$. Each node $k \in \mathcal{B}_{lq}$ then updates

$$\mathbf{V}^{i+1}_k = \mathbf{V}^i_k \mathbf{G}_l \tag{84}$$

$$\mathbf{W}^{i+1}_k = \mathbf{W}^i_k \mathbf{H}_l \,. \tag{85}$$

5) $i \leftarrow i+1$ and $q \leftarrow (q \mod K) + 1$.
6) Return to step 2.

where $\overline{\mathbf{x}}^i_{\rightarrow k}$ is the stacked version of all the signals $\overline{\mathbf{x}}^i_{qk}$ for $q \in \mathcal{N}_k$, where the $\overline{\mathbf{x}}^i_{qk}$'s are ordered such that $\overline{\mathbf{x}}_{mk}$ is above $\overline{\mathbf{x}}_{lk}$ if $m < l$. In the sequel, we will always use the same order whenever we stack variables that relate to the different neighbors of node k. We similarly define $\widetilde{\mathbf{y}}^i_k$ as

$$\widetilde{\mathbf{y}}^i_k \triangleq \begin{bmatrix} \mathbf{y}_k \\ \overline{\mathbf{y}}^i_{\rightarrow k} \end{bmatrix} \,. \tag{87}$$

We also define $\mathcal{B}_{kq}$ as the set of nodes in the tree branch that would be disconnected from the rest of the tree if the link between nodes k and q is cut, and where $k \in \mathcal{B}_{kq}$ and $q \notin \mathcal{B}_{kq}$. It is noted that, by resolving the dependencies between the $\overline{\mathbf{x}}^i_{kq}$'s in (78), we find that

$$\overline{\mathbf{x}}^i_{kq} = \sum_{l\in\mathcal{B}_{kq}} \mathbf{V}^{i\,T}_l \mathbf{x}^i_l \tag{88}$$

and similarly,

$$\overline{\mathbf{y}}^i_{kq} = \sum_{l\in\mathcal{B}_{kq}} \mathbf{W}^{i\,T}_l \mathbf{y}^i_l \,. \tag{89}$$

Example: Consider the network graph depicted in Fig. 3, and consider nodes 3 and 4 in particular. It holds that $\mathcal{B}_{34} = \{1,2,3\}$ and $\mathcal{B}_{43} = \{4,5,6,7,8,9\}$. The fused observations that node 3 transmits to node 4 correspond to (see (78))

$$\overline{\mathbf{x}}^i_{34} = \mathbf{V}^{i\,T}_3 \mathbf{x}_3 + \overline{\mathbf{x}}^i_{13} + \overline{\mathbf{x}}^i_{23} \,.$$

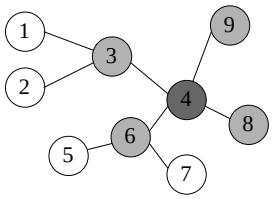

Fig. 3. Example of a network graph with tree topology with 9 sensor nodes.

The fused signals that node 4 receives from all its neighbors in $\mathcal{N}_4$ are stacked in

$$\overline{\mathbf{x}}^i_{\to 4} = \begin{bmatrix} \overline{\mathbf{x}}^i_{34} \\ \overline{\mathbf{x}}^i_{64} \\ \overline{\mathbf{x}}^i_{84} \\ \overline{\mathbf{x}}^i_{94} \end{bmatrix} .$$

It is also noted that (see (88))

$$\overline{\mathbf{x}}^i_{34} = \sum_{j=1}^{3} \mathbf{V}_j^{i\,T} \mathbf{x}_j \text{ , and } \overline{\mathbf{x}}^i_{43} = \sum_{j=4}^{9} \mathbf{V}_j^{i\,T} \mathbf{x}_j \text{ .}$$

The description of the DCCA algorithm for tree topology networks is given in Table II. Similar to the DCCA algorithm in fully-connected networks, the transmission of the $\mathbf{G}$ and $\mathbf{H}$ parameters is assumed to be negligible compared to the transmission of the signal observations between nodes, assuming $L \gg Q^2$ (see also Remark IV.1). We also re-iterate that the communication efficiency can be significantly improved by using local broadcasts and RFC signals instead of the TFC signals defined in (78).

The convergence analysis of the DCCA algorithm for fully-connected networks in Section IV-B can be generalized to the case of tree topology networks, although this will require some modification in the proofs and in the notation (details omitted).

Remark VI.1. It is noted that, in a context of BSS, a node k can reconstruct the Q source signals extracted by the Q principal CCA directions by computing (compare with (71))

$$\overline{\mathbf{s}} = \lim_{i \to \infty} \left(\overline{\mathbf{x}}^i_k + \sum_{q \in \mathcal{N}_k} \overline{\mathbf{x}}^i_{qk} \right) . \tag{90}$$

Remark VI.2. The DCCA algorithm typically converges slower in a WSN with a tree topology compared to a fully-connected WSN. This is due to the smaller number of degrees of freedom with which a node can update the different submatrices in $\mathbf{V}^i$ and $\mathbf{W}^i$, i.e., P is smaller in Table II compared to Table I.

VII. Numerical simulations

In this section, we provide Monte-Carlo (MC) simulations of the DCCA algorithm in a BSS scenario.

A. Generation of sensor signal observations

All plots in this section show the averaged results over 300 MC runs (unless stated otherwise). In each MC run, a new scenario is created with K nodes (we run simulations for different network sizes K), each collecting observations of a different M_k-dimensional stochastic vector $\mathbf{x}_k$, where $M_k = 6$, $\forall\, k \in \mathcal{K}$, unless stated otherwise. The observations of the stacked vector $\mathbf{x}$ are generated as

$$\mathbf{x}[t] = \mathbf{A}\mathbf{s}[t] + \mathbf{n}[t] \tag{91}$$

where $\mathbf{A}$ is a deterministic $6K \times 10$ mixing matrix (independent of t), $\mathbf{s}$ is a vector containing 10 source signals, and $\mathbf{n}$ models spatially uncorrelated noise. In each MC run, the entries of $\mathbf{A}$ are randomly drawn from a uniform distribution over the interval $[-0.5; 0.5]$. The noise term $\mathbf{n}$ is generated by a temporally white stochastic process that is uniformly distributed over the interval $[-\sqrt{0.25}/2; \sqrt{0.25}/2]$. The 10 source signals in $\mathbf{s}$ are created as filtered versions of a temporally white stochastic process that is uniformly distributed over the interval $[-0.5; 0.5]$. The first source is convolved with the filter vector $\mathbf{1}_{20}$, where $\mathbf{1}_J$ denotes a J-dimensional all-ones vector. The second source is convolved with $\mathbf{1}_{18}$, the third with $\mathbf{1}_{16}$, and so on. This guarantees that each source has a different autocorrelation structure, which is one of the requirements for CCA-based BSS. Finally, all sources are scaled to unity power.

B. Simulation results

We apply CCA-based BSS using three time lags, i.e., $D = 3$ in (64). In this case $\hat{\mathbf{V}}$ serves as a demixing matrix for the first Q sources of $\mathbf{s}$, which can be extracted by computing $\overline{\mathbf{s}}[t] = \hat{\mathbf{V}}^T \mathbf{x}[t]$. In the case of DCCA, the demixing matrix $\mathbf{V}^i$ changes in each iteration, and therefore we write $\overline{\mathbf{s}}^i[t] = \mathbf{V}^{i\,T} \mathbf{x}[t]$. To assess the performance of the DCCA algorithm, we use the signal-to-error power ratio (SER) (in dB) for the extraction of the first source (after compensating for the scaling ambiguity[8]), i.e.,

$$\text{SER}^i = 10 \log_{10} \frac{E\{s_1^2\}}{E\{(\overline{s}_1^i - s_1)^2\}} . \tag{92}$$

Note that the higher the SER, the better the performance of the source extraction. We also assess the performance by means of the mean squared error (MSE) between the coefficients of the obtained demixing matrix and those of the demixing matrix of the centralized CCA algorithm (after resolving the sign ambiguity), i.e.,

$$\text{MSE}^i = \frac{1}{MQ} \|\mathbf{V}^i - \hat{\mathbf{V}}\|_F^2 . \tag{93}$$

The upper plot in Fig. 4 shows the SER for 50 iterations of the DCCA algorithm in a network of $K = 10$ sensor nodes, for different values of Q, i.e., $Q = 1$, $Q = 2$, and $Q = 3$. The plot demonstrates the convergence of the DCCA algorithm to the optimal SER, i.e., the SER obtained by the centralized CCA algorithm. The bottom plot shows the median (and the 25% and 75% percentiles) of the MSE over the different MC runs, demonstrating that the entries of $\mathbf{V}^i$ converge to the entries of $\hat{\mathbf{V}}$. It is observed that the convergence improves when a larger Q is used, which is due to the extra degrees of freedom in each updating step (at the cost of a larger bit rate over the wireless links). It is noted that the MSE will theoretically decrease to 0, i.e., to the machine precision in practice.

[8]The scaling ambiguity is resolved by computing the optimal scaling factor between the two signals $\overline{s}_1^i$ and s_1 under a least squares criterion.

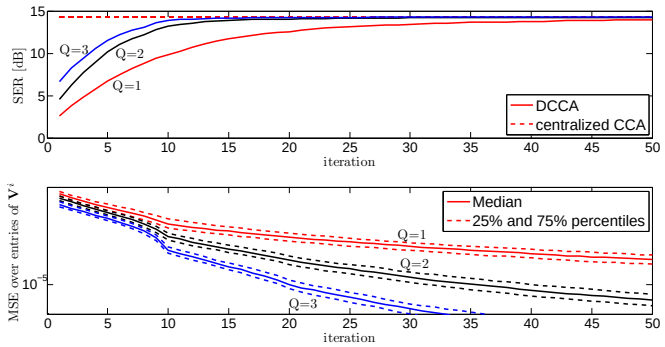

Fig. 4. Convergence properties of the DCCA algorithm in a fully-connected WSN with $K = 10$ nodes, for different values of Q.

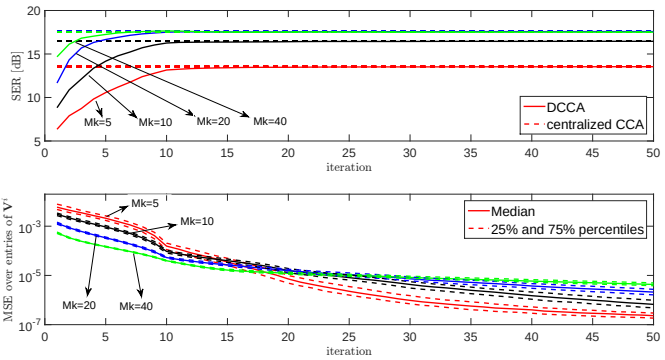

Fig. 5. Convergence properties of the DCCA algorithm in a fully-connected WSN with $K = 10$ nodes, for different values of M_k ($Q = 3$).

Fig. 5 demonstrates how the convergence speed is influenced by the number of sensors per node or per cluster, i.e., for different values of M_k (fixing $Q = 3$, $K = 10$). Although the SER measure is not heavily influenced by the value of M_k, the MSE plot shows an increased convergence speed for lower values of M_k.

Fig. 6 shows the first 800 observations of the original source signals s_j, $j = 1, \ldots, Q$, and the estimated source signals $\overline{s}_j^i$, $j = 1, \ldots, Q$, after $i = 50$ iterations of the DCCA algorithm (for $Q = 3$) and the centralized CCA algorithm (for a single MC run). The residual mismatch between the estimated source signals and the original source signals is due to the noise component $\mathbf{n}$ in (91), which is not incorporated in the signal model of CCA-based BSS.

Fig. 7, shows the convergence results for $Q = 1$ and for different network sizes, i.e., $K = 10$, $K = 20$ and $K = 40$. It is observed that the network size does not have a major impact on the convergence of the DCCA algorithm.

Fig. 8 shows the convergence results of the DCCA algorithm when applied in a tree topology for different values of K (the MMSE curve for $K = 20$ has been omitted for intelligibility purposes). In each MC run, a different random tree is generated. It is again observed that the DCCA algorithm converges to the centralized solution, albeit slower than in a fully-connected network (see also Remark VI.2).

Finally, we study the convergence properties in a finite-window simulation, where the theoretical convergence analysis is only approximately satisfied due to the appearance of estimation errors in the correlation matrices[9]. In Fig. 9, we

[9]In the other simulations, the second-order statistics were estimated over the full signal length (both in the distributed and centralized case).

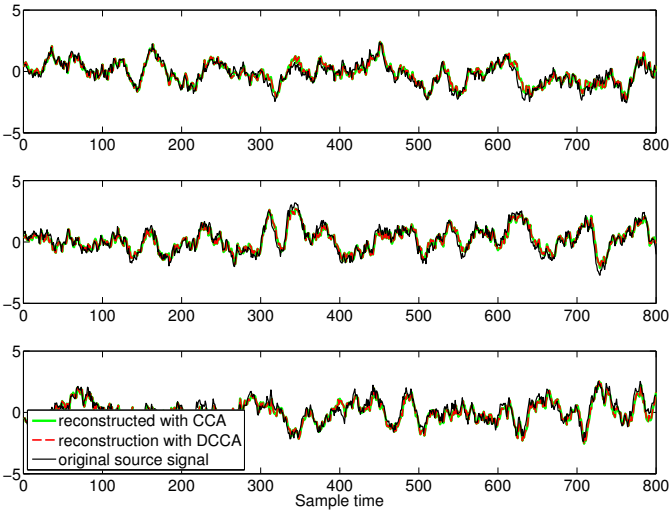

Fig. 6. The first $Q = 3$ source signals and the corresponding reconstructed signals using the DCCA-based demixing matrix $\mathbf{V}^i$ after $i = 50$ iterations.

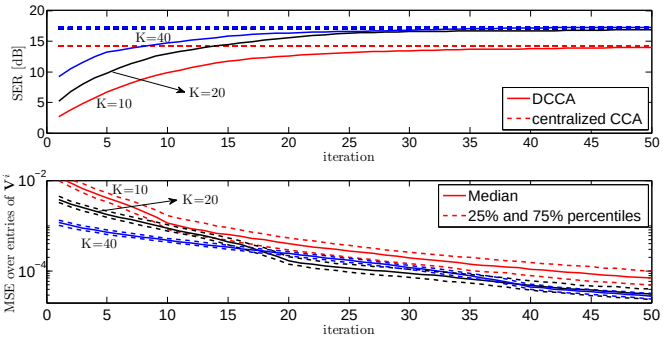

Fig. 7. Convergence properties of the DCCA algorithm with $Q = 1$ in fully-connected WSNs with different network size K.

show the performance of the centralized and distributed CCA algorithm where the second order statistics are estimated over a window of L samples, which shifts over time in each iteration (see Remark IV.1). Although both algorithms show a drop in performance depending on the window size L, we see that the discrepancy between the centralized and distributed algorithm is negligible.

VIII. Conclusions

We have proposed a time-recursive distributed CCA (DCCA) algorithm, which allows to estimate and track Q principal CCA directions from the sensor signal observations collected by the different nodes in a wireless sensor network with a fully-connected or a tree topology. The DCCA algorithm is able to iteratively estimate these network-wide principal CCA directions without explicitly constructing the corresponding network-wide correlation matrices. Instead, each node broadcasts Q-dimensional compressed observations of its sensor signals, and solves local CCA problems with a smaller dimension. This significantly reduces the bit rate over the wireless links as well as the computational load compared to the centralized implementation (at the cost of a slower tracking). We have proven convergence of the DCCA algorithm to the network-wide CCA directions, and we have demonstrated its performance by means of numerical simulations in a blind source separation scenario.

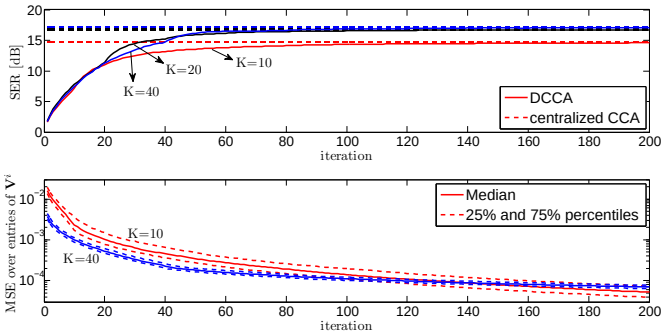

Fig. 8. Convergence properties of the DCCA algorithm with $Q = 1$ in WSNs with a tree topology for different network sizes K.

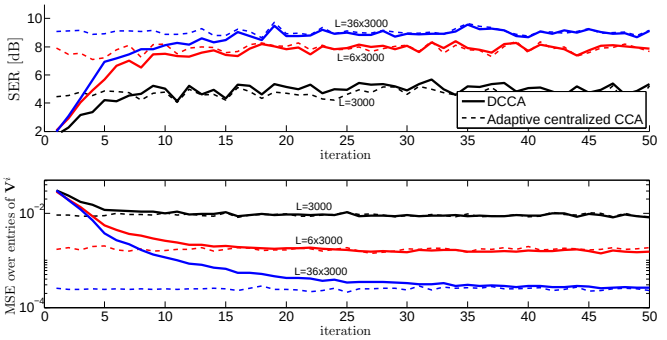

Fig. 9. Performance of the centralized and DCCA algorithms where the correlation matrices are estimated over a window of L samples (fully-connected, $K = 5$, $Q = 2$). The MSE is with respect to batch-mode CCA over the full signal.

APPENDIX

A. Proof of Result IV.2

From (9)-(10), we find that $\hat{\mathbf{V}}$ and $\hat{\mathbf{W}}$ can also be jointly computed from a larger GEVD

$$\begin{bmatrix} \mathbf{O} & \mathbf{R}_{xy} \\ \mathbf{R}_{yx} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{V}} \\ \hat{\mathbf{W}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{xx} & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_{yy} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{V}} \\ \hat{\mathbf{W}} \end{bmatrix} \mathbf{\Sigma} \,. \tag{94}$$

We will use this expression as a necessary and sufficient condition for the Q principal CCA directions at a later stage in this proof.

Assume w.l.o.g. that node q is the updating node at iteration i of the DCCA algorithm. Since $\widetilde{\mathbf{V}}_q^{i+1}$ and $\widetilde{\mathbf{W}}_q^{i+1}$, as defined in Table I, contain the Q principal CCA directions corresponding to $\widetilde{\mathbf{x}}_q^i$ and $\widetilde{\mathbf{y}}_q^i$, we have that (see (9)-(10))

$$\mathbf{R}^i_{\widetilde{x}_q\widetilde{y}_q} \widetilde{\mathbf{W}}_q^{i+1} = \mathbf{R}^i_{\widetilde{x}_q\widetilde{x}_q} \widetilde{\mathbf{V}}_q^{i+1} \widetilde{\mathbf{\Sigma}}_q^i \tag{95}$$

and

$$\mathbf{R}^i_{\widetilde{y}_q\widetilde{x}_q} \widetilde{\mathbf{V}}_q^{i+1} = \mathbf{R}^i_{\widetilde{y}_q\widetilde{y}_q} \widetilde{\mathbf{W}}_q^{i+1} \widetilde{\mathbf{\Sigma}}_q^i \tag{96}$$

where $\widetilde{\mathbf{\Sigma}}_q^i$ is a diagonal matrix containing the Q largest canonical correlation coefficients between $\widetilde{\mathbf{x}}_q^i$ and $\widetilde{\mathbf{y}}_q^i$. We now only proceed with (95), but note that a similar strategy can be followed for (96). With (42)-(44), we find that (95) can be written as

$$\mathcal{V}_q^{i\,T} \mathbf{R}_{xy} \mathcal{W}_q^i \widetilde{\mathbf{W}}_q^{i+1} = \mathcal{V}_q^{i\,T} \mathbf{R}_{xx} \mathcal{V}_q^i \widetilde{\mathbf{V}}_q^{i+1} \widetilde{\mathbf{\Sigma}}_q^i \,. \tag{97}$$

We now assume that $(\mathbf{V}^i, \mathbf{W}^i) \in \mathcal{E}$, i.e., $(\mathbf{V}^i, \mathbf{W}^i)$ is an equilibrium point. This means that $(\mathbf{V}^{i+1}, \mathbf{W}^{i+1}) = (\mathbf{V}^i, \mathbf{W}^i)$, and from (50)-(57) it follows that $\widetilde{\mathbf{V}}_q^{i+1} = \left[\mathbf{V}_q^{i\,T}\ \mathbf{I}_Q\ \ldots\ \mathbf{I}_Q\right]^T$ and $\widetilde{\mathbf{W}}_q^{i+1} = \left[\mathbf{W}_q^{i\,T}\ \mathbf{I}_Q\ \ldots\ \mathbf{I}_Q\right]^T$. Therefore, and using the notation (27)-(29), in an equilibrium point we have that

$$\mathcal{V}_q^{i\,T} \mathbf{R}_{xy} \mathbf{W}^i = \mathcal{V}_q^{i\,T} \mathbf{R}_{xx} \mathbf{V}^i \widetilde{\mathbf{\Sigma}}_q^i \,. \tag{98}$$

Selecting the first M_q rows in (98), and relying on (27), yields

$$\mathbf{U}_q \mathbf{R}_{xy} \mathbf{W}^i = \mathbf{U}_q \mathbf{R}_{xx} \mathbf{V}^i \widetilde{\mathbf{\Sigma}}_q^i \tag{99}$$

where

$$\mathbf{U}_q \triangleq \left[\mathbf{O}_{M_q \times \underline{M}_q}\ \mathbf{I}_{M_q}\ \mathbf{O}_{M_q \times \overline{M}_q}\right] \tag{100}$$

with $\underline{M}_q \triangleq \sum_{j=1}^{q-1} M_j$, and $\overline{M}_q \triangleq \sum_{j=q+1}^{K} M_j$. Furthermore, after left-multiplying (98) with the matrix $\left[\mathbf{V}_q^{i\,T}\ \mathbf{I}_Q\ \ldots\ \mathbf{I}_Q\right]$, and relying on the fact that $\mathbf{V}^{i\,T} \mathbf{R}_{xx} \mathbf{V}^i = \mathbf{I}_Q$, $\forall\, i \in \mathbb{N}_0$ (see Result IV.1), we obtain

$$\mathbf{V}^{i\,T} \mathbf{R}_{xy} \mathbf{W}^i = \widetilde{\mathbf{\Sigma}}_q^i \,. \tag{101}$$

In case of an equilibrium point, $\mathbf{V}^{i+1} = \mathbf{V}^i$ for all updating nodes $q \in \mathcal{K}$, hence the same reasoning can be performed for all $q \in \mathcal{K}$. Using this result, and by stacking the K matrix equations as defined in (99), $\forall\, q \in \mathcal{K}$, we obtain

$$\mathbf{R}_{xy} \mathbf{W}^i = \begin{bmatrix} \mathbf{U}_1 \mathbf{R}_{xx} \mathbf{V}^i \widetilde{\mathbf{\Sigma}}_1^i \\ \vdots \\ \mathbf{U}_K \mathbf{R}_{xx} \mathbf{V}^i \widetilde{\mathbf{\Sigma}}_K^i \end{bmatrix} \,. \tag{102}$$

Furthermore, since the lefthand side of (101) is independent of q, this shows that $\widetilde{\mathbf{\Sigma}}_q^i = \widetilde{\mathbf{\Sigma}}_k^i = \widetilde{\mathbf{\Sigma}}^i$, $\forall\, k, q \in \mathcal{K}$. Therefore, (102) is equal to

$$\mathbf{R}_{xy} \mathbf{W}^i = \mathbf{R}_{xx} \mathbf{V}^i \widetilde{\mathbf{\Sigma}}^i \,. \tag{103}$$

Starting from (96), and using a similar reasoning, we also find that

$$\mathbf{R}_{yx} \mathbf{V}^i = \mathbf{R}_{yy} \mathbf{W}^i \widetilde{\mathbf{\Sigma}}^i \,. \tag{104}$$

Remember that $\widetilde{\mathbf{\Sigma}}^i$ is a diagonal matrix, and therefore (103)-(104) shows that $(\mathbf{V}^i, \mathbf{W}^i)$ is a solution of the GEVD (94), and hence the columns of $\mathbf{V}^i$ and $\mathbf{W}^i$ must correspond to CCA directions of $\mathbf{x}$ and $\mathbf{y}$ (although not necessarily the Q principal directions). Therefore, and since we have assumed that $(\mathbf{V}^i, \mathbf{W}^i)$ is an arbitrary equilibrium point in $\mathcal{E}$, we conclude that any equilibrium point can only contain CCA directions of $\mathbf{x}$ and $\mathbf{y}$. Note that (103)-(104) defines a necessary condition for $(\mathbf{V}^i, \mathbf{W}^i)$ to be an equilibrium point, but not a sufficient condition.

The fact that $(\hat{\mathbf{V}}, \hat{\mathbf{W}}) \in \mathcal{E}$ follows straightforwardly from the fact that $(\hat{\mathbf{V}}, \hat{\mathbf{W}})$ maximizes (15)-(17), and the fact that the DCCA algorithm results in a monotonic increase of $f(\mathbf{V}, \mathbf{W})$ under the constraints (16)-(17) (Result IV.1). Note that (61) also assures that (15)-(17) has a unique maximum[10].

Finally, we have to prove that $(\hat{\mathbf{V}}, \hat{\mathbf{W}})$ is the only stable equilibrium point. An equilibrium point $(\mathbf{V}^*, \mathbf{W}^*)$ is stable under the DCCA update rules if and only if any infinitesimal perturbation that does not result in a violation of the constraints

[10]Even if $(\hat{\mathbf{V}}, \hat{\mathbf{W}})$ is not unique, the fix in Appendix C will ensure that $(\hat{\mathbf{V}}, \hat{\mathbf{W}}) \in \mathcal{E}$, i.e., $(\hat{\mathbf{V}}, \hat{\mathbf{W}})$ does not change under the DCCA updates.

(16)-(17) does not increase $f(\mathbf{V}^*, \mathbf{W}^*)$, i.e.,

$$\begin{aligned} &\exists\, \delta > 0,\ \forall\, (\Delta\mathbf{V}, \Delta\mathbf{W}) \in \mathbb{U}: \\ &\|\Delta\mathbf{V}\|_F + \|\Delta\mathbf{W}\|_F \leq \delta \Rightarrow \\ &\quad f(\mathbf{V}^* + \Delta\mathbf{V}, \mathbf{W}^* + \Delta\mathbf{W}) \leq f(\mathbf{V}^*, \mathbf{W}^*) \end{aligned} \tag{105}$$

where $\mathbb{U}$ is the set of possible perturbations such that the constraints (16)-(17) are not violated when $\mathbf{V} = \mathbf{V}^* + \Delta\mathbf{V}$ and $\mathbf{W} = \mathbf{W}^* + \Delta\mathbf{W}$. Indeed, since f is monotonically increasing under the DCCA updates (Result IV.1), equilibrium points $(\mathbf{V}^*, \mathbf{W}^*)$ that do not satisfy (105) are unstable under the DCCA update rules since a small perturbation may cause $f(\mathbf{V}^* + \Delta\mathbf{V}, \mathbf{W}^* + \Delta\mathbf{W}) \geq f(\mathbf{V}^*, \mathbf{W}^*)$, and since $f\left(\mathbf{V}^{i+1}, \mathbf{W}^{i+1}\right) \geq f\left(\mathbf{V}^i, \mathbf{W}^i\right)$ for all subsequent iterations $i \in \mathbb{N}$, the DCCA algorithm cannot return to the original equilibrium point $(\mathbf{V}^*, \mathbf{W}^*)$. If $(\mathbf{V}^*, \mathbf{W}^*)$ contains CCA directions that are not in $(\hat{\mathbf{V}}, \hat{\mathbf{W}})$, it does not satisfy (105), since perturbations in the direction of the principal CCA components will increase the objective function $f(\mathbf{V}, \mathbf{W})$. Therefore, $(\hat{\mathbf{V}}, \hat{\mathbf{W}})$ is the only point that both satisfies the equilibrium conditions (103)-(104) and the stability condition (105), and hence it is the only stable equilibrium point.

B. Proof of Result IV.3

Since $f(\mathbf{V}^i, \mathbf{W}^i)$ increases monotonically (Result IV.1), and since it has an upper bound, we have that

$$\lim_{i\to\infty} \left(f(\mathbf{V}^{i+1}, \mathbf{W}^{i+1}) - f(\mathbf{V}^i, \mathbf{W}^i)\right) = 0\,. \tag{106}$$

Because $(\widetilde{\mathbf{V}}^{i+1}, \widetilde{\mathbf{W}}^{i+1})$ defines the Q principal CCA directions for $\widetilde{\mathbf{x}}_q^i$ and $\widetilde{\mathbf{y}}_q^i$, it is also the global maximum of (45)-(47). Because of (61), we know that this global maximum is unique. Therefore, and due to the continuity of (45)-(47) and its equivalence with the constrained optimization problem (19)-(23), we know that

$$\begin{aligned} &\forall\, \delta > 0, \exists\, \mu > 0,\ \forall\, \mathbf{V}, \mathbf{W} \in \mathcal{C}: \\ &\quad |f(\mathbf{V}^{i+1}, \mathbf{W}^{i+1}) - f(\mathbf{V}, \mathbf{W})| < \mu \Rightarrow \\ &\quad\ \|\mathbf{V}^{i+1} - \mathbf{V}\|_F + \|\mathbf{W}^{i+1} - \mathbf{W}\|_F < \delta \end{aligned} \tag{107}$$

where $\mathcal{C}$ denotes the constraint set of the optimization problem (19)-(23). Together with (106), this implies that

$$\lim_{i\to\infty} \left(\|\mathbf{V}^{i+1} - \mathbf{V}^i\|_F + \|\mathbf{W}^{i+1} - \mathbf{W}^i\|_F\right) = 0\,. \tag{108}$$

We now use this result to prove convergence of the DCCA algorithm.

The proof of Result IV.2 relies on the fact that $(\mathbf{V}^i, \mathbf{W}^i) \in \mathcal{E}$, which is used to obtain (98) from (97). However, if $(\mathbf{V}^i, \mathbf{W}^i) \notin \mathcal{E}$, then $(\mathbf{V}^{i+1}, \mathbf{W}^{i+1}) \neq (\mathbf{V}^i, \mathbf{W}^i)$ and therefore an error term $\mathbf{E}_q^i$ should be added in (98), i.e.,

$$\mathcal{V}_q^{i\,T}\mathbf{R}_{xy}\mathbf{W}^i = \mathcal{V}_q^{i\,T}\mathbf{R}_{xx}\mathbf{V}^i\widetilde{\mathbf{\Sigma}}_q^i + \mathbf{E}_q^i\,. \tag{109}$$

Therefore, an error term then also appears in (101) and (102), which are derived from (98), i.e.,

$$\mathbf{R}_{xy}\mathbf{W}^i + \mathbf{\Delta}^i = \begin{bmatrix} \mathbf{U}_1\mathbf{R}_{xx}\mathbf{V}^i\widetilde{\mathbf{\Sigma}}_1^i \\ \vdots \\ \mathbf{U}_K\mathbf{R}_{xx}\mathbf{V}^i\widetilde{\mathbf{\Sigma}}_K^i \end{bmatrix}\,. \tag{110}$$

and

$$\mathbf{V}^{i\,T}\mathbf{R}_{xy}\mathbf{W}^i + \mathbf{\Delta}_q^i = \widetilde{\mathbf{\Sigma}}_q^i \tag{111}$$

where $\mathbf{\Delta}^i$ and $\mathbf{\Delta}_q^i$, $\forall\, q \in \mathcal{K}$, are error terms. However, from (108), it follows that the error $\mathbf{E}_q^i$ vanishes in (109) if $i \to \infty$, and therefore

$$\lim_{i\to\infty} \|\mathbf{V}^{i\,T}\mathbf{R}_{xy}\mathbf{W}^i - \widetilde{\mathbf{\Sigma}}_k^i\|_F = 0\,,\ \forall\, k \in \mathcal{K} \tag{112}$$

$$\lim_{i\to\infty} \|\mathbf{R}_{xy}\mathbf{W}^i - \mathbf{R}_{xx}\mathbf{V}^i\widetilde{\mathbf{\Sigma}}^i\|_F = 0 \tag{113}$$

where

$$\widetilde{\mathbf{\Sigma}}^i = \mathbf{V}^{i\,T}\mathbf{R}_{xy}\mathbf{W}^i\,. \tag{114}$$

Using a similar strategy, we also find from (104) that

$$\lim_{i\to\infty} \|\mathbf{R}_{yx}\mathbf{V}^i - \mathbf{R}_{yy}\mathbf{W}^i\widetilde{\mathbf{\Sigma}}^i\|_F = 0\,. \tag{115}$$

Since $\widetilde{\mathbf{\Sigma}}_k^i$, $\forall\, k \in \mathcal{K}$, are diagonal matrices (by construction), it follows from (112) and (114) that $\lim_{i\to\infty} \widetilde{\mathbf{\Sigma}}^i$ is also diagonal. This fact, together with (113) and (115) shows that $(\mathbf{V}^i, \mathbf{W}^i)$ converges to CCA directions of $\mathbf{x}$ and $\mathbf{y}$ when $i \to \infty$. Furthermore, it follows from (108) that the columns of $\mathbf{V}^i$ and $\mathbf{W}^i$ cannot switch between different CCA directions over the different iterations if $i \to \infty$, which proves the result.

C. Algorithm fixes for special cases

1) Rank deficient $\mathbf{R}^i_{\tilde{x}_q\tilde{x}_q}$ and/or $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$: In the rare case where $\mathbf{R}^i_{\tilde{x}_q\tilde{x}_q}$ and/or $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$ is rank deficient, then the local CCA solution in iteration i at node q is ill-defined. Rank deficiency of these matrices occurs when there is a node k for which $\mathbf{V}_k^i$ and/or $\mathbf{W}_k^i$ has linearly dependent columns. This problem can be circumvented by letting node k replace the linearly dependent column in $\mathbf{V}_k^i$ or $\mathbf{W}_k^i$ by random entries, yielding a new $\overline{\mathbf{x}}_k^i$ or $\overline{\mathbf{y}}_k^i$ in which all channels are linearly independent. Note that a linearly dependent column is always redundant, and hence its removal and/or replacement can not counteract the monotonic increase of $f(\mathbf{V}_k^i, \mathbf{W}_k^i)$.

2) Degenerated canonical correlation coefficients: In the rare case where the n-th largest canonical correlation coefficient of $\widetilde{\mathbf{x}}_k^i$ and $\widetilde{\mathbf{y}}_k^i$ is degenerate, i.e. $\widetilde{\rho}_n^i = \widetilde{\rho}_{n+1}^i$ (with $n \leq Q$), then $\widetilde{\mathbf{V}}_q^{i+1}$ and $\widetilde{\mathbf{W}}_q^{i+1}$ become ill-defined in their n-th and $(n+1)$-th column, as there are multiple solutions. One pragmatic fix is to skip node q in the current update round, assuming that the problem will disappear in the next update round. If the problem persists, to ensure convergence one should select the solution of the degenerate CCA problem that is closest to the solution from the previous iteration.

REFERENCES

[1] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, pp. 321–377, 1936.

[2] M. Borga, *Learning Multidimensional Signal Processing*. PhD thesis, Linköping University, SE-581 83 Linköping, Sweden, 1998.

[3] L. Sun, S. Ji, S. Yu, and J. Ye, "On the equivalence between canonical correlation analysis and orthonormalized partial least squares," in *Proceedings of the 21st international jont conference on Artifical intelligence*, ser. IJCAI'09. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009, pp. 1230–1235.

[4] J. Kettenring, "Canonical analysis of several sets of variables," *Biometrika*, vol. 58, no. 3, pp. 433–451, Dec. 1971.

[5] Y.-O. Li, T. Adali, W. Wang, and V. Calhoun, "Joint blind source separation by multiset canonical correlation analysis," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3918–3929, 2009.
[6] O. Friman, M. Borga, P. Lundberg, and H. Knutsson, "Exploratory fMRI analysis by autocorrelation maximization," *NeuroImage*, vol. 16, pp. 454–464, June 2002.
[7] S. Assecondi, H. Hallez, S. Staelens, A. M. Bianchi, G. M. Huiskamp, and I. Lemahieu, "Removal of the ballistocardiographic artifact from EEG-fMRI data: a canonical correlation approach," *Physics in Medicine and Biology*, vol. 54, no. 6, pp. 1673–1689, 2009.
[8] N. Correa, Y.-O. Li, T. Adali, and V. Calhoun, "Canonical correlation analysis for feature-based fusion of biomedical imaging modalities and its application to detection of associative networks in schizophrenia," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 6, pp. 998–1007, 2008.
[9] N. Correa, T. Adali, Y.-O. Li, and V. Calhoun, "Canonical correlation analysis for data fusion and group inferences," *IEEE Signal Processing Magazine*, vol. 27, no. 4, pp. 39–50, 2010.
[10] W. De Clercq, A. Vergult, B. Vanrumste, W. Van Paesschen, and S. Van Huffel, "Canonical correlation analysis applied to remove muscle artifacts from the electroencephalogram," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 12, pp. 2583–2587, 2006.
[11] P. Gannabathula and I. S. N. Murthy, "ARMA model estimation for EEG using canonical correlation analysis," in *Engineering in Medicine and Biology Society, 1988. Proceedings of the Annual International Conference of the IEEE*, 1988, pp. 1204–1205 vol.3.
[12] J. Xie, T. Qiu, and L. Wen-hong, "An ocular artifacts removal method based on canonical correlation analysis and two-channel eeg recordings," in *World Congress on Medical Physics and Biomedical Engineering May 26-31, 2012, Beijing, China*, ser. IFMBE Proceedings, M. Long, Ed. Springer Berlin Heidelberg, 2013, vol. 39, pp. 465–468.
[13] K. Ciftci, B. Sankur, A. Akin, and Y. P. Kahya, "Processing near infrared spectroscopy signals using canonical correlation analysis," in *IEEE Signal Processing and Communications Applications*, 2006.
[14] H. Ge, I. Kirsteins, and X. Wang, "Does canonical correlation analysis provide reliable information on data correlation in array processing?" in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 2113–2116.
[15] Z. Bai, G. Huang, and L. Yang, "A radar anti-jamming technology based on canonical correlation analysis," in *International Conference on Neural Networks and Brain (ICNN B).*, vol. 1, 2005, pp. 9–12.
[16] M. Sargin, Y. Yemez, E. Erzin, and A. Tekalp, "Audiovisual synchronization and fusion using canonical correlation analysis," *IEEE Transactions on Multimedia*, vol. 9, no. 7, pp. 1396–1403, 2007.
[17] A. Dogandzic and A. Nehorai, "Finite-length MIMO equalization using canonical correlation analysis," *IEEE Transactions on Signal Processing*, vol. 50, no. 4, pp. 984–989, 2002.
[18] J. Via and I. Santamaria, "Adaptive blind equalization of SIMO systems based on canonical correlation analysis," in *IEEE Workshop on Signal Processing Advances in Wireless Communications*, 2005, pp. 318–322.
[19] K. Todros and A. Hero, "On measure transformed canonical correlation analysis," *IEEE Trans. Signal Proc.*, vol. 60, no. 9, pp. 4570–4585, 2012.
[20] M. Borga and H. Knutsson, "A canonical correlation approach to blind source separation," *Dept. Biomedical Engineering, Linköping University, Linköping, Sweden, Tech. Rep. LiU-IMT-EX-0062*, 2001.
[21] W. Liu, D. Mandic, and A. Cichocki, "Analysis and online realization of the CCA approach for blind source separation," *IEEE Transactions on Neural Networks*, vol. 18, no. 5, pp. 1505–1510, 2007.
[22] ——, "Blind source separation based on generalised canonical correlation analysis and its adaptive realization," in *Proc. Congress on Image and Signal Processing (CISP)*, vol. 5, 2008, pp. 417–421.
[23] S. Schell and W. Gardner, "Programmable canonical correlation analysis: a flexible framework for blind adaptive spatial filtering," *IEEE Transactions on Signal Processing*, vol. 43, no. 12, pp. 2898–2908, 1995.
[24] A. Yeredor, "Performance analysis of GEVD-based source separation with second-order statistics," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 5077–5082, 2011.
[25] A. M. Tom, "The generalized eigendecomposition approach to the blind source separation problem," *Digital Signal Processing*, vol. 16, no. 3, pp. 288 – 302, 2006.
[26] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 434–444, 1997.
[27] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*. John Wiley & Sons, 2001.
[28] B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester, "A survey on wireless body area networks," *Wireless Networks*, vol. 17, no. 1, pp. 1–18, Jan. 2011.
[29] A. Bertrand, "Distributed signal processing for wireless EEG sensor networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2015.
[30] A. Bertrand and M. Moonen, "Distributed eye blink artifact removal in a wireless EEG sensor network," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP)*, Florence, Italy, May 2014.
[31] Y. Chi, S. Deiss, and G. Cauwenberghs, "Non-contact low power EEG/ECG electrode for high density wearable biopotential sensor networks," in *International Workshop on Wearable and Implantable Body Sensor Networks*, 2009, pp. 246–250.
[32] A. Bertrand and M. Moonen, "Distributed adaptive estimation of covariance matrix eigenvectors in wireless sensor networks with application to distributed PCA," *Signal Processing*, vol. 104, pp. 120–135, 2014.
[33] ——, "Distributed adaptive estimation of node-specific signals in wireless sensor networks with a tree topology," *IEEE Trans. Signal Processing*, vol. 59, no. 5, pp. 2196–2210, May 2011.
[34] K. Muller, "Understanding canonical correlation through the general linear model and principal components," *The American Statistician*, vol. 36, no. 4, pp. 342–354, 1982.
[35] S. Leurgans, R. Moyeed, and B. Silverman, "Canonical correlation analysis when the data are curves," *Journal of the Royal Statistical Society*, vol. 55, no. 3, pp. 725–740, 1993.
[36] H. Vinod, "Canonical ridge and econometrics of joint production," *J. Econometrics*, vol. 4, pp. 147–166, 1976.
[37] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. Baltimore: The Johns Hopkins University Press, 1996.

Alexander Bertrand (M'08) received the Ph.D. degree in Engineering Sciences (Electrical Engineering) from KU Leuven, Belgium, in 2011. He is currently assistant professor with the Electrical Engineering Department, KU Leuven, and principal investigator with iMinds Medical IT. He was a visiting researcher at University of California, Los Angeles, at Imec-NL Eindhoven (The Netherlands), and at University of California, Berkeley. Dr. Bertrand received the 2012 FWO/IBM Award, the 2013 KU Leuven Research Council Award, and an Honorary Mention in the 2013 ERCIM Cor Baayen Award. He has served as a technical program committee member for EUSIPCO and IEEE GlobalSIP, and as lead guest editor for Signal Processing. His main research focus is in biomedical signal processing and distributed signal estimation.

Marc Moonen (M'94, SM'06, F'07) is a Full Professor at the E.E. Department of KU Leuven, where he is heading a research team working in the area of numerical algorithms and signal processing for digital communications, wireless communications, DSL and audio signal processing. He received the 1994 KU Leuven Research Council Award, the 1997 Alcatel Bell (Belgium) Award (with Piet Vandaele), the 2004 Alcatel Bell (Belgium) Award (with Raphael Cendrillon), and was a 1997 Laureate of the Belgium Royal Academy of Science. He received journal best paper awards from the IEEE Transactions on Signal Processing and from Elsevier Signal Processing. He was chairman of the IEEE Benelux Signal Processing Chapter (1998- 2002), and a member of the IEEE Signal Processing Society Technical Committee on Signal Processing for Communications. He was President of EURASIP (2007-2007, 2011-2012) and is currently Past-President of EURASIP. He has served as Editor-in-Chief for the EURASIP Journal on Applied Signal Processing (2003-2005), Area Editor for Feature Articles in IEEE Signal Processing Magazine (2012-2014) and has been a member of the editorial board of IEEE Transactions on Circuits and Systems II, IEEE Signal Processing Magazine, Integration-the VLSI Journal, EURASIP Journal on Wireless Communications and Networking, and Signal Processing. He is currently a member of the editorial board of EURASIP Journal on Advances in Signal Processing.